

2

# **HAZARDOUS MATERIAL LIFE-CYCLE COST MODEL**

## **TECHNICAL MANUAL**

**VERSION 1.0**

**DTIC**  
**ELECTE**  
**JAN 8 1993**  
**S C D**

**H. L. Ly**

**D. M. Pearsall**

**93-00491**



**93 1 07 051**

**Report No. 92-19**

Approved for public release: distribution unlimited.

**NAVAL HEALTH RESEARCH CENTER  
P.O. BOX 85122  
SAN DIEGO, CALIFORNIA 92186-5122**

**NAVAL MEDICAL RESEARCH AND DEVELOPMENT COMMAND  
BETHESDA, MARYLAND**



**Hazardous Material Life-Cycle Cost Model**  
**Technical Manual**  
**Version 1.0**

Prepared by:

Hoa Le Ly  
 Dianna M. Pearsall

Naval Health Research Center  
 Medical Information Systems and  
 Operations Research Department  
 P.O. Box 85122  
 San Diego, CA 92186-5122

Accession For	
NTIS	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

DTIC QUALITY INSPECTED 3

Report No. TR-92-19 was supported by the Naval Medical Research and Development Command, Bethesda, MD, Department of the Navy, under a NAVSUP Reimbursable Work Unit. The views expressed in this article are those of the authors and do not reflect the official policy or position of the Department of the Navy, Department of Defense, nor the U.S. Government.

---

## **SUMMARY**

This technical manual contains the information on the program source code, data elements, file structures needed to maintain the Hazardous Material Life-Cycle Cost Module. This documentation was created using the FOXDOC Version 2.0 program.

---

## TABLE OF CONTENTS

Introduction . . . . .	1
Section I System Summary . . . . .	1
Section II Menu Summary . . . . .	1
Section III Screen Summary . . . . .	3
Section IV Data Dictionary . . . . .	11
A. Database Structure Summary . . . . .	11
B. Database Field Summary . . . . .	16
Section V Tree Diagram . . . . .	18
Section VI Procedure and Function Summary . . . . .	29
Section VII Program Source Code . . . . .	49

**Introduction.** This document was created using the FOXDOC Version 2.0 program to generate the technical documentation for the Hazardous Material Life-Cycle Cost Module (HMLCCM) system. The documentation is separated into seven sections (1) System Summary, (2) Menu Summary, (3) Screen Summary, (4) Data Dictionary, (5) Tree Diagram, (6) Procedure Summary, and (7) Source Code Program Listings.

**Section I. System Summary.** See Section V for the tree diagram of the programs, procedures, functions and format files.

The HMLCCM system has:

```

11087 lines of code
  66 program files
  30 procedure files
 155 procedures and functions
   1 menu file
  10 screen files
 944 cross-referenced tokens
  16 databases
  10 structural index files

```

**Section II. Menu Summary.** The following description lists the menu options used to drive the system. HMENU.MNX is the only menu template created.

<u>OPTION NAME</u>	<u>KEYS\FUNCTION CALLED</u>
System	ALT+S _MSM_SYSTM
Help...	F1 _MST_HELP
-----	_MST_SP100
Calculator	_MST_CALCUL
-----	
HMLCCM	(Submenu HMLCCM)
Cost Analysis	(Submenu COSTANALYS)
Build Hazmat Scenario	do hmsc
Reference Material	do hmref
System Maintenance	(Submenu SYSTEMMAIN)
Back-Up (floppy)	(Procedure)
Up-Load Data	(Procedure)
Set Parameters	(Submenu SETPARAMET)
Materials	CTRL+M do hmat.spr
Life Cycle Phase	CTRL+L do hmlc.spr
Process	CTRL+W do hmwp.spr
EXposure Type	CTRL+X do hmet.spr
Cost Factors	CTRL+F do hmcfe.spr
Cost Factor Elements	CTRL+E do hmcfe.spr
Cost Factor Items	CTRL+I do hmcfei.spr
Build Hazmat Table	CTRL+B do hmtab.spr
File	ALT+F _MSM_FILE
Printer Setup...	_MFI_SETUP
Print...	_MFI_PRINT
Quit	do _quit in hminit

<u>OPTION NAME</u>	<u>KEYS\FUNCTION CALLED</u>
Edit	ALT+E    MSM_EDIT
Undo	CTRL+U    MED_UNDO
Redo	CTRL+R    MED_REDO
Cut	CTRL+X    MED_CUT
Copy	CTRL+C    MED_COPY
Paste	CTRL+V    MED_PASTE
Clear	MED_CLEAR
-----	MED_SP200
Select All	CTRL+A    MED_SLCTA
-----	MED_SP300
Goto Line...	MED_GOTO
Find...	CTRL+F    MED_FIND
Find Again	CTRL+G    MED_FINDA
Replace And Find Again	CTRL+E    MED_REPL
Replace All	MED_REPLA
-----	MED_SP400
Preferences...	MED_PREF
Database	ALT+D    MSM_DATA
Browse	MDA_BROW
-----	MDA_SP100
Sort...	MDA_SORT
Total...	MDA_TOTAL
-----	MDA_SP200
Average...	MDA_AVG
Count...	MDA_COUNT
Sum...	MDA_SUM
Calculate...	MDA_CALC
Report...	MDA_REPRT
Record	ALT+R    MSM_REC RD
Goto...	MRC_GOTO
Locate...	MRC_LOCAT
Continue	CTRL+K    MRC_CONT
Seek...	MRC_SEEK
-----	MRC_SP200
Replace...	MRC_REPL
Delete...	MRC_DELET
Recall...	MRC_RECAL
Program	ALT+P    MSM_PROG
Cancel	MPR_CANCL
Resume	CTRL+M    MPR_RESUM
Window	ALT+W    MSM_WINDO
Clear	MWI_CLEAR

Section III. Screen Summary. Ten (10) screen files were used as input templates for the system; HMAT.SCX, HMLC.SCX, HMWP.SCX, HMET.SCX, HMC.F.SCX, HMC.FE.SCX, HMC.FEI.SCX, HMTAB.SCX, HMCOMP.SCX, and W\_PRINT.SCX.

HMAT.SCX

Last updated: 09/24/92 at 8:14

#### HAZARDOUS MATERIALS

```

0
1  ID 2:      NAME 3: hmatname.....
2
3
4
5  Synonyms/Trade Names  Common Uses
6
7  Description            Chemical/Physical Properties
8
9  Occupational           Exposure Exposure Limits
10
11 Health Hazards         Medical Surveillance
12
13 Special Tests          Treatment
14
15 All Reference          Personal Protective Equipment
16
17  <Save>  <Cancel>

```

< Add >  
 < Edit >  
 <Previous>  
 < Next >  
 < E\<xit>

Window name: Hmat

Coordinates: FROM 0,0 TO 0,77

Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: m.Action	Push button	"@*VN \\<Add;\\<Edit;\\!\\<Next;\\<Previous;\\?E\\<xit"
2: M.hmatid	Field	
3: M.hmatname	Field	"@!"
4: m.Save	Push button	"@*VN \\<Save;\\<Cancel"

HMLC.SCX

Last updated: 09/24/92 at 8:14

```

0          HM LIFE CYCLE PHASES
1
2          PHASE ID: 2: hmlcid.
3
4          PHASE NAME: 3: hmlc.....
5
6
7
8          < Save > <Cancel>
9
          < Add >
          < Edit >
          * \<Next *
          <Previous>
          < E\<xit >

```

Window name: Hmlc  
 Coordinates: FROM 0,0 TO 0,51  
 Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: m.Action	Push button	"@*VN \<Add;\<Edit;\\<Next;\<Previous;\?E\<xit"
2: m.hmlcid	Field	
3: m.hmlc	Field	"@!"
4: m.Save	Push button	"@*HN \<Save;\<Cancel"

HMWP.SCX

Last updated: 10/09/92 at 8:14

```

0          HM PROCESSES
1
2          ID NUM: 2: hmwpid.
3
4          PROCESS: 3: hmw.....
5          .....
6          .....
7          .....
8          .....
9          .....
10
11          < Save > <Cancel>
12
          < Add >
          < Edit >
          * Next *
          <Previous>
          < Exit >

```

Window name: Hmwp  
 Coordinates: FROM 0,0 TO 0,63  
 Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: m.Action	Push button	"@*VN \<Add;\<Edit;\\<Next;\<Previous;\?E\<xit"
2: m.hmwpid	Field	
3: m.hmwp	Field	"@!"
4: m.Save	Push button	"@*HN \<Save;\<Cancel"



HMET.SCX

Last updated: 09/24/92 at 8:14

```

0          HM EXPOSURE TYPES
1
2          ID NUM: 3: hmetid.
3
4          TYPE: 4: hmet.....
5
6
7
8          < Save > <Cancel>
9
          < Add >
          < Edit >
          « \<Next »
          <Previous>
          < E\<xit >

```

Window name: Hmet

Coordinates: FROM 0,0 TO 0,51

Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: m.Action	Push button	"@*VN \<Add;\<Edit;\\<Next;\<Previous;\?E\<xit"
2: m.Save	Push button	"@*HN \<Save;\<Cancel"
3: m.hmetid	Field	
4: m.hmet	Field	"@!"

HMCF.SCX

Last updated: 09/24/92 at 8:14

```

0          HM COST FACTORS
1
2          ID NUM: 2: hmcfid.
3
4          COST FACTOR: 3: hmcfc.....
5
6
7
8          < Save > <Cancel>
9
          < Add >
          < Edit >
          « \<Next »
          <Previous>
          < E\<xit >

```

Window name: Hmcf

Coordinates: FROM 0,0 TO 0,52

Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: m.Action	Push button	"@*VN \<Add;\<Edit;\\<Next;\<Previous;\?E\<xit"
2: m.hmcfid	Field	
3: m.hmcf	Field	"@!"
4: m.Save	Push button	"@*HN \<Save;\<Cancel"

HMCFE.SCX

Last updated: 09/24/92 at 8:14

```

0          HM COST FACTOR ELEMENTS
1
2          < Add  >
3
4          < Edit  >
5          FACTOR:  2: hm 3: answr..... < Next  >
6
7          ELEMENT: 4: hm 5: hmcfe..... <Previous>
8
9          " \<Top  >
10
11          < Save > <Cancel>
12          < Bottom >
13          < E\<xit >

```

Window name: Hmcfe  
Coordinates: FROM 0,0 TO 0,66  
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: m.Action	Push button	"@*VN \\<Add;\\<Edit;\\<Next;\\<Previous;\\!\\<Top;\\<Bottom;\\?E\\<xit"
2: m.hmcfid	Field	
3: m.answr	Field	
4: m.hmcfeid	Field	
5: m.hmcfe	Field	"@!"
6: m.Save	Push button	"@*HN \\<Save;\\<Cancel"

```

0          COST FACTOR ELEMENT ITEMS          < Add >
1
2          < Edit>
3          FACTOR: 2: hm  3: answr.....
4          < Next >
5          ELEMENT: 4: hm  5: hmcfe.....
6          <Previous>
7          ITEM: 6: hm  7: hmcfei.....
8          " \<Top>
9          COST:  8: hmcfeic
10         < Bottom>
11
12         < Save > <Cancel>
13
14         < Browse>
15         < Exit >

```

Window name: Hmcfel

Coordinates: FROM 0,0 TO 0,68

Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: m.Action	Push button	"@*VN \<Add;\<Edit;\<Next;\<Previous; \! \<Top;\<Bottom;B\<rowse;\?E\<xit"
2: m.hmcfid	Field	
3: m.answr	Field	
4: m.hmcfeld	Field	
5: m.hmcfe	Field	"@!"
6: M.hmcfelid	Field	
7: M.hmcfel	Field	"@!"
8: M.hmcfelcost	Field	"@\$"
9: m.Save	Push button	"@*HN \<Save;\<Cancel"

HMTAB.SCX

Last updated: 09/24/92 at 8:14

```

0      #2: t                      HAZARDOUS MATERIALS TABLE
1      Material: 3: h 4: hmname.....
2      ..... Per: 5: hmunit.....
3
4      Phase: 6: h 7: hmlc.....
5-6    Process: 8: hm 9: hmwp.....
7      Exposure Type:10: 11: hmet.....
8      Probability of exposure:....12: hmetp
9
10     Factor: 13: 14: hmcfe.....
11     Element: 15: 16: hmcfe.....
12     Item: 17: 18: hmcfei.....
14     Cost: 19: hmcfe Probability 20: p
15         By Person ( ) YES(=) NO
16         By Day ( ) YES(=) NO
17         By Quantity ( ) YES(=) NO
18
19     < Save > < Cancel >

```

Window name: Hmtab

Coordinates: FROM 0,0 TO 0,78

Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: m.Action	Push button	"@*VN
<Add;\<Edit;\!\\<Next;\<Previous;\<Top;\<Bottom;\?E\<xit		
2: m.tabid	Field	
3: m.hmatid	Field	"@z"
4: m.hmname	Field	"@!"
5: m.hmunit	Popup	"@^"
6: m.hmlcid	Field	"@z"
7: m.hmlc	Field	"@!"
8: m.hmwpid	Field	"@z"
9: m.hmwp	Field	"@!"
10: m.hmetid	Field	"@z"
11: m.hmet	Field	"@!"
12: m.hmetprob	Field	
13: m.hmcfid	Field	"@z"
14: m.hmcf	Field	
15: m.hmcfeid	Field	"@z"
16: m.hmcfe	Field	"@!"
17: m.hmcfeid	Field	"@z"
18: m.hmcfei	Field	"@!"
19: m.hmcfecost	Field	"@\$\$\$,\$\$\$.\$99"
20: m.prob	Field	"9.99"
21: m.perp	Radio button	"@*RHN YES;NO"
22: m.perd	Radio button	"@*RHN YES;NO"
23: m.perq	Radio button	"@*RHN YES;NO"
24: m.Save	Push button	"@*VN \<Save;\<Cancel"

HMCOMP.SCX

Last updated: 10/13/92 at 8:14

COMPUTE COST VALUE

0	Estimated	Scenario:1: 2: chmscname...	Estimated
1	Cost		Variance
2			
3	[ ] for Step:	4 step .....	[ ]
4			
5			
6	[ ] for Factor:	7: fact.....	[ ]
7			
8			
9	[ ] for Factor	10: wfact.. at step 11: wstep..	[ ]
10			
11			
12	[ ] for Phase:	14: phase.....	[ ]
13			
14			
15	# of Iterations:	Total of Scenario cost: ( ) Yes (*) No	
16	16: sample..		
17		< Ok > <Browse> <Cancel>	

Window name: W hmcost

Coordinates: FROM 0,0 TO 0,76

Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: m.chmscid	Field	"@Z"
2: m.chmscname	Field	"@!T"
3: m.cstep	Check box	"@*C "
4: m.step	Popup	"@^ "
5: m.bstep	check box	"@*C "
6: m.cfact	Check box	"@*C "
7: m.fact	Popup	"@^ "
8: m.bfact	Check box	"@*C "
9: m.cwstep	Check box	"@*C "
10: m.wfact	Popup	"@^ "
11: m.wstep	Popup	"@^ "
12: m.bwstep	Check box	"@*C "
13: m.cphase	Check box	"@*C "
14: m.phase	Popup	"@^ "
15: m.bphase	Check box	"@*C "
16: m.sample	Field	"@Z"
17: m.scost	Radio button	"@*RHN Yes ;No"
18: m.action	Push button	"@*HN\<Ok;\<Browse;\<Cancel"

---

W\_PRINT.SCX

Last updated: 09/24/92 at 8:14

Print Option

```
0
1 Print to:.      (=) File
2                ( ) Printer
3
4
5
6 Filename: 1: mprintfile.....
7
8
9
10      < OK >  <Cancel>
```

Window name: W\_prn

Coordinates: FROM 0,0 TO 0,42

Window options: FLOAT CLOSE MINIMIZE SHADOW

<u>Name</u>	<u>Type</u>	<u>Picture</u>
1: mprintfile	Field	
2: mbuttons	Push button	"@*HT OK;\?Cancel"
3: mchoice	Radio button	"@*RVN File;Printer"

Section IV. Data Dictionary. There are sixteen database files in the HMLCCM system:

HMAT.DBF	-- Hazardous Materials File
HMLC.DBF	-- Hazardous Material Life Cycle Table
HMWP.DBF	-- Hazardous Material Work Process Table
HMET.DBF	-- Hazardous Material Environment Type Table
HMCF.DBF	-- Hazardous Material Cost Factor Table
HMCFE.DBF	-- Hazardous Material Cost Factor Element Table
HMCFEI.DBF	-- Hazardous Material Cost Factor Element Item Table
HMUNIT.DBF	-- Hazardous Material Unit Table
HMTAB.DBF	-- Hazardous Material Scenario Table
HMSTEP.DBF	-- Hazardous Material Step Table
HMCOMP.DBF	-- Hazardous Material Cost Computation File
HMSCEN.DBF	-- Hazardous Material Scenario File
BOOT.DBF	-- Temporary file used for the bootstrap computations
HMTEMP.DBF	-- Temporary file used to store data from tables
TEMP.DBF	-- Temporary file used to store data from tables
TEST.DBF	-- Temporary file used to store data from tables

#### A. Database Structure Summary.

Structure for database : HMAT.DBF

Field	Field name	Type	Width	Dec	Start	End
1	HMATID	Numeric	4		1	4
2	HMATNAME	Character	45		5	49
3	MFG	Character	25		50	74
4	NIIN	Character	9		75	83
5	MILSPEC	Character	11		84	94
6	HMCOST	Numeric	10	2	95	104
7	HMUNIT	Character	10		105	114
8	HMDESCRIPTION	Memo	10		115	124
9	CHEM_PHY	Memo	10		125	134
10	HEALTH_HAZ	Memo	10		135	144
11	MED_SURV	Memo	10		145	154
12	OCC_EXP	Memo	10		155	164
13	PPE_TREAT	Memo	10		165	174
14	SPEC_TESTS	Memo	10		175	184
15	TREATMENT	Memo	10		185	194
16	SYN_TRADE	Memo	10		195	204
17	COM_USES	Memo	10		205	214
18	EXP_LIMITS	Character	10		215	224

\*\* Total \*\*

225

This database is associated with the memo file: HMAT.FPT

This database appears to be associated with multiple index file(s):

C:\HMLCCM\DATA\HMAT.CDX

Used by: HMAT.SPR

: GET_HMID()	(function in HMTAB.SPR)
: GET_HMNAME()	(function in HMTAB.SPR)
: GET_HMAT()	(function in HMSTEP.PRG)
: GET_HMATN()	(function in HMSTEP.PRG)

---

Structure for database : HMLC.DBF

Field	Field name	Type	Width	Dec	Start	End
1	HMLCID	Numeric	2		1	2
2	HMLC	Character	30		3	32
** Total **			33			

This database appears to be associated with multiple index file(s):  
C:\HMLCCM\DATA\HMLC.CDX :

Used by: HMLC.SPR  
: GET\_HMLCN() (function in HMTAB.SPR)  
: Q9B0S1X10() (function in HMTAB.SPR)  
: GET\_HMLC() (function in HMTAB.SPR)

Structure for database : HMWP.DBF

Field	Field name	Type	Width	Dec	Start	End
1	HMWPID	Numeric	2		1	2
2	HMWP	Character	80		3	82
3	HMPLATFORM	Numeric	2		83	84
** Total **			85			

This database appears to be associated with multiple index file(s):  
C:\HMLCCM\DATA\HMWP.CDX :

Used by: HMWP.SPR  
: GET\_HMWPN() (function in HMTAB.SPR)  
: Q9B0S1X20() (function in HMTAB.SPR)  
: GET\_HMWP() (function in HMTAB.SPR)

Structure for database : HMET.DBF

Field	Field name	Type	Width	Dec	Start	End
1	HMETID	Numeric	2		1	2
2	HMET	Character	30		3	32
** Total **			33			

This database appears to be associated with multiple index file(s):  
C:\HMLCCM\DATA\HMET.CDX :

Used by: HMET.SPR  
: GET\_HMETN() (function in HMTAB.SPR)  
: Q9B0S1YHS() (function in HMTAB.SPR)  
: GET\_HMET() (function in HMTAB.SPR)  
: GET\_HMETID() (function in HMTAB.SPR)



---

Structure for database : HMC.F.DBF            Alias: CFTMP

Field	Field name	Type	Width	Dec	Start	End
1	HMC.FID	Numeric	2		1	2
2	HMC.F	Character	30		3	32
** Total **			33			

This database appears to be associated with multiple index file(s):  
C:\HMLCCM\DATA\HMC.F.CDX :

Used by: HMC.F.SPR

: GET_HMC.F()	(function in HMC.FE.SPR)
: GET_CF	(procedure in HMC.FE.SPR)
: GET_CFID	(procedure in HMC.FE.SPR)
: REL	(procedure in HMC.FE.SPR)
: BROWSEITEM	(procedure in HMC.FEI.SPR)
: GET CFAR()	(function in HMTAB.SPR)
: Q9B0S1Z0J()	(function in HMTAB.SPR)
: GET_HMC.FID()	(function in HMTAB.SPR)

Structure for database : HMC.FEI.DBF            Alias: CFEITMP

Field	Field name	Type	Width	Dec	Start	End
1	HMC.FEIID	Numeric	4		1	4
2	HMC.FEID	Numeric	2		5	6
3	HMC.FID	Numeric	2		7	8
4	HMC.FEI	Character	30		9	38
5	HMC.FEINO	Character	15		39	53
6	HMC.FEICOST	Numeric	10	2	54	63
7	HMC.FEIUNIT	Character	10		64	73
** Total **			74			

This database appears to be associated with multiple index file(s):  
C:\HMLCCM\DATA\HMC.FEI.CDX :

Used by: REL

: BROWSEITEM	(procedure in HMC.FE.SPR)
: GET_EI()	(procedure in HMC.FEI.SPR)
: GET_EIID()	(function in HMTAB.SPR)
	(function in HMTAB.SPR)

---

Structure for database : HMCFE.DBF                      Alias: CFETMP

Field	Field name	Type	Width	Dec	Start	End
1	HMCFEID	Numeric	2		1	2
2	HMCFID	Numeric	2		3	4
3	HMCFE	Character	30		5	34
** Total **			35			

This database appears to be associated with multiple index file(s):                      :  
C:\HMLCCM\DATA\HMCFE.CDX

Used by: REL                      (procedure in HMCFE.SPR)  
         : GET\_HMCFE()              (function in HMCFEI.SPR)  
         : GET\_CFE                  (procedure in HMCFEI.SPR)  
         : GET\_CFEID                (procedure in HMCFEI.SPR)  
         : BROWSEITEM               (procedure in HMCFEI.SPR)  
         : GET\_CFAR1()              (function in HMTAB.SPR)  
         : GET\_CF1()                (function in HMTAB.SPR)  
         : GET\_EID()                (function in HMTAB.SPR)

Structure for database : HMUNIT.DBF

Field	Field name	Type	Width	Dec	Start	End
1	HMUNIT	Character	20		1	20
** Total **			21			

This database appears to be associated with multiple index file(s):                      :  
C:\HMLCCM\DATA\HMUNIT.CDX

Used by: HMTAB.SPR

Structure for database : HMTAB.DBF                      Alias: HMTAB

Field	Field name	Type	Width	Dec	Start	End
1	TABID	Numeric	4		1	4
2	HMATID	Numeric	4		5	8
3	HMLCID	Numeric	2		9	10
4	HMWPID	Numeric	2		11	12
5	HMETID	Numeric	2		13	14
6	HMETPROB	Float	5		15	19
7	HMCFID	Numeric	2		20	21
8	HMCFEID	Numeric	2		22	23
9	HMCFEIID	Numeric	2		24	25
10	PERP	Numeric	1		26	26
11	PERD	Numeric	1		27	27
12	PERQ	Numeric	1		28	28
13	HMCFEFCOST	Numeric	5	2	29	33
14	HMUNIT	Character	10		34	43
15	WTAVERAGE	Float	5		44	48
16	PROB	Float	5		49	53
** Total **			54			

Used by: HMTAB.SPR                      (procedure in HMTAB.SPR)  
         : CHANGE                      (procedure in HMTAB.SPR)  
         : GET\_TABLE                  (procedure in HMTAB.SPR)  
         : SELWT()                    (function in HMTAB.SPR)  
         : GET\_HMUNIT                (procedure in HMTAB.SPR)

---

Structure for database : HMCOMP.DBF Status: Temporary

Used by: HMCOMP.SPR  
: COMPUTSTEP() (function in HMCOMP.SPR)  
: COMPUTFACT() (function in HMCOMP.SPR)  
: COMPUTWFACT() (function in HMCOMP.SPR)  
: COMPUTPHASE() (function in HMCOMP.SPR)  
: BSTEP() (function in HMCOMP.SPR)  
: BFACT() (function in HMCOMP.SPR)  
: BWSTEP() (function in HMCOMP.SPR)  
: BPHASE() (function in HMCOMP.SPR)

Structure for database : HMSTEP.DBF

Field	Field name	Type	Width	Dec	Start	End
1	HMSCID	Numeric	4		1	4
2	HMSTEP	Numeric	4		5	8
3	HMATID	Numeric	4		9	12
4	HMLCID	Numeric	2		13	14
5	HMWPID	Numeric	2		15	16
6	PERNUM	Numeric	3		17	19
7	DURNUM	Numeric	3		20	22
8	QTYNUM	Numeric	3		23	25
9	WSTNUM	Numeric	3		26	28
10	UNIT	Character	20		29	48
** Total **			49			

Used by: GET HMARRAY (procedure in HMCOMP.SPR)  
: HMSTEP.PRG

Structure for database : TEMP.DBF Status: Temporary File

Used by: GET\_TABLE (procedure in HMCOMP.SPR)

Structure for database : TEST.DBF Status: Temporary File

Used by: CALCULATE() (function in HMCOMP.SPR)

Structure for database: BOOT.DBF Status: Temporary File

Used by: BSTEP() (function in HMCOMP.SPR)  
: BFACT() (function in HMCOMP.SPR)  
: BWSTEP() (function in HMCOMP.SPR)  
: BPHASE() (function in HMCOMP.SPR)

Structure for database : HMSCEN.DBF

Field	Field name	Type	Width	Dec	Start	End
1	HMSCID	Numeric	4		1	4
2	HMSCNAME	Character	40		5	44
** Total **			45			

This database appears to be associated with multiple index file(s):  
C:\HMLCCM\DATA\HMSCEN.CDX :

Used by: HMSCEN.PRG  
: GET\_HMSC() (function in HMSCEN.PRG)  
: SCSAVE (procedure in HMSTEP.PRG)

---

Structure for database : HMTEMP.DBF

Field	Field name	Type	Width	Dec	Start	End
1	HMSCID	Numeric	4		1	4
2	HMSTEP	Numeric	4		5	8
3	HMATID	Numeric	4		9	12
4	HMLCID	Numeric	2		13	14
5	HMWPID	Numeric	2		15	16
6	PERNUM	Numeric	3		17	19
7	DURNUM	Numeric	3		20	22
8	QTYNUM	Numeric	3		23	25
9	WSTNUM	Numeric	3		26	28
10	UNIT	Character	20		29	48
** Total **			49			

Used by: SCSAVE

(procedure in HMSTEP.PRG)

B. Database Field Summary

Field Name	Type	Len	Dec	Database
CHEM PHY	M	10	0	HMAT.DBF
COM USES	M	10	0	HMAT.DBF
DURNUM	N	3	0	HMTEMP.DBF
				HMSTEP.DBF
EXP LIMITS	C	10	0	HMAT.DBF
HEALTH HAZ	M	10	0	HMAT.DBF
HMATID	N	4	0	HMTEMP.DBF
				HMTAB.DBF
				HMSTEP.DBF
				HMAT.DBF
HMATNAME	C	45	0	HMAT.DBF
HMCF	C	30	0	HMCF.DBF
HMCFE	C	30	0	HMCFE.DBF
HMCFE COST	N	5	2	HMTAB.DBF
HMCFEI	C	30	0	HMCFEI.DBF
HMCFEI COST	N	10	2	HMCFEI.DBF
HMCFEID	N	2	0	HMCFEI.DBF
				HMCFE.DBF
				HMTAB.DBF
HMCFEIID	N	2	0	HMTAB.DBF
HMCFEIID	N	4	0	HMCFEI.DBF
HMCFEINO	C	15	0	HMCFEI.DBF
HMCFEIUNIT	C	10	0	HMCFEI.DBF
HMCFID	N	2	0	HMCF.DBF
				HMCFEI.DBF
				HMTAB.DBF
				HMCFE.DBF
HMCOST	N	10	2	HMAT.DBF
HMDESCRIPT	M	10	0	HMAT.DBF
HMET	C	30	0	HMET.DBF
HMETID	N	2	0	HMTAB.DBF
				HMET.DBF
HMETPROB	F	5	2	HMTAB.DBF
HMLC	C	30	0	HMLC.DBF
HMLCID	N	2	0	HMTAB.DBF
				HMLC.DBF
				HMTEMP.DBF
				HMSTEP.DBF

---

HMPLATFORM	N	2	0	HMWP.DBF
HMSCID	N	4	0	HMTEMP.DBF
				HMSTEP.DBF
				HMSCEN.DBF
HMSCNAME	C	40	0	HMSCEN.DBF
HMSTEP	N	4	0	HMTEMP.DBF
				HMSTEP.DBF
HMUNIT	C	10	0	HMTAB.DBF
				HMAT.DBF
HMUNIT	C	20	0	HMUNIT.DBF
HMWP	C	80	0	HMWP.DBF
HMWPID	N	2	0	HMTEMP.DBF
				HMTAB.DBF
				HMWP.DBF
				HMSTEP.DBF
MED_SURV	M	10	0	HMAT.DBF
MFG_	C	25	0	HMAT.DBF
MILSPEC	C	11	0	HMAT.DBF
NIIN	C	9	0	HMAT.DBF
OCC EXP	M	10	0	HMAT.DBF
PERD	N	1	0	HMTAB.DBF
PERNUM	N	3	0	HMSTEP.DBF
				HMTEMP.DBF
PERP	N	1	0	HMTAB.DBF
PERQ	N	1	0	HMTAB.DBF
PPE TREAT	M	10	0	HMAT.DBF
PROB	F	5	2	HMTAB.DBF
QTYNUM	N	3	0	HMSTEP.DBF
				HMTEMP.DBF
SPEC TESTS	M	10	0	HMAT.DBF
SYN TRADE	M	10	0	HMAT.DBF
TABID	N	4	0	HMTAB.DBF
TREATMENT	M	10	0	HMAT.DBF
UNIT	C	20	0	HMSTEP.DBF
				HMTEMP.DBF
WSTNUM	N	3	0	HMSTEP.DBF
				HMTEMP.DBF
WTAVERAGE	F	5	2	HMTAB.DBF

**Section V. Tree Diagram.** The tree diagram lists each program that is called by the order in which it is used. Under each program there is a list of all functions called and the where the procedure/function is stored.

#### HMINIT.PRG

- SET() (Function in FP)
- UPPER() (FP function)
- HMGETPWD() (function in HMGETPWD.PRG)
- IIF() (FP function)
- BIGCHARS() (function in BIGCHARS.PRG)
- HMENU.MPR

#### HMREF.PRG

- SET() ... (FP function)
- OPENFILE() (function in HMSC.PRG)
  - FILEFIND() (function in HMSC.PRG)
    - PARAMETER() (FP function)
    - FILE() (FP function)
    - DP() (function in DP.PRG)
    - LOCFILE() (FP function)
  - USED() (FP function)
- HMLU() (function in HMLU.PRG)
- EMPTY() (FP function)
- HMVIEW() (function in HMREF.PRG)
  - PARAMETER() ... (FP function)
  - FOUND() (FP function)
  - CHR() (FP function)
  - NEWLINE() (function in HMREF.PRG)
    - CHR() ... (FP function)
    - EMPTY() ... (FP function)
  - IIF() ... (FP function)
  - EMPTY() ... (FP function)
  - UPPER() ... (FP function)
  - ALLTRIM() (FP function)
  - MEMOWIN() (function in MEMOWIN.PRG)

#### HMSC.PRG

- SET() ... (FP function)
- HMSCEN() (function in HMSCEN.PRG)
- DP() ... (function in DP.PRG)
- INT() (FP function)
- VAL() (FP function)
- EMPTY() ... (FP function)
- HMSTEP() (function in HMSTEP.PRG)
- DEL\_HMSC() (function in HMSC.PRG)
  - OPENFILE() ... (function in HMSC.PRG)
- YESNO() (function in YESNO.PRG)

#### HMCOMP.SPR

- HMCOMP.DBF (database)
- PARAMETER() ... (FP function)
- SET() ... (FP function)
- WEXIST() (FP function)
- INT() ... (FP function)
- SROW() (FP function)
- SCOL() (FP function)
- ON() (FP function)
- OPEN (procedure in HMCOMP.SPR)
  - OPENFILE() ... (function in HMSC.PRG)
  - CANCEL (procedure in HMCOMP.SPR)
    - ERRMSG() (function in ERRMSG.PRG)
    - CANCEL... (procedure in HMCOMP.SPR)

```

GET_HMARRAY      (procedure in HMCOMP.SPR)
  HMSTEP.DBF (database)
  SELECT() (FP function)
  RECCOUNT() (FP function)
  STR() (FP function)
  RECNO() (FP function)
GET_TABLE        (procedure in HMCOMP.SPR)
  HMTAB.DBF (database)
  TEMP.DBF (database)
  SELECT() ... (FP function)
ALLTRIM() ... (FP function)
ASCAN() (FP function)
AINS() (FP function)
WVISIBLE() (FP function)
_Q9BOS2GG5() (HMCOMP.SPR)
_Q9BOS2GVA() (HMCOMP.SPR)
_Q9BOS2H5I() (HMCOMP.SPR)
_Q9BOS2HGT() (HMCOMP.SPR)
_Q9BOS2HR2() (HMCOMP.SPR)
_Q9BOS2HYD() (HMCOMP.SPR)
SETUPBOOT() (HMCOMP.SPR)
  ACOPY() (FP function)
  ADEL() (FP function)
  ALEN() (FP function)
  BSELECT() (function in BSELECT.PRG)
  EMPTY() ... (FP function)
  ERRMSG() ... (function in ERRMSG.PRG)
  ASCAN() ... (FP function)
  ASUBSCRIPT() (FP function)
  BWFACT() (function in BWFACT.PRG)
  DP() ... (function in DP.PRG)
POPUPSHOW() (function in HMSC.PRG)
  WEXIST() ... (FP function)
  INT() ... (FP function)
  SROW() ... (FP function)
  SCOL() ... (FP function)
  WVISIBLE() ... (FP function)
COMPUTE          (procedure in HMCOMP.SPR)
  SELECT() ... (FP function)
  RECCOUNT() ... (FP function)
  COMPUTSTEP() (function in HMCOMP.SPR)
    HMCOMP.DBF (database)
    SELECT() ... (FP function)
    CALCULATE() (function in HMCOMP)
      TEST.DBF (database)
      SELECT() .. (FP function)
      USED() ... (FP function)
      RECCOUNT() .. (FP function)
      EMPTY() .. (FP function)
      IIF() ... (FP function)
      RECNO() .. (FP function)
      DELFILE() .. (Func.in HMCOMP)
        USED() (FP function)
  ALLTRIM() ... (FP function)
  STR() ... (FP function)
  EMPTY() ... (FP function)
  ALEN() ... (FP function)
  COMPUTFACT() (function in HMCOMP.SPR)
    HMCOMP.DBF (database)

```

	PARAMETER() ... (FP function)
	SELECT() ... (FP function)
	CALCULATE() (function in HMCOMP)
COMPUTWFACT()	(function in HMCOMP)
	HMCOMP.DBF (database)
	PARAMETER() ... (FP function)
	SELECT() ... (FP function)
	IIF() ... (FP function)
	EMPTY() ... (FP function)
	ALLTRIM() ... (FP function)
	STR() ... (FP function)
	CALCULATE() (function in HMCOMP)
COMPUTPHASE()	(HMCOMP.SPR)
	HMCOMP.DBF (database)
	PARAMETER() ... (FP function)
	SELECT() ... (FP function)
	CALCULATE() (function in HMCOMP)
CHR()	... (FP function)
BSTEP()	(HMCOMP.SPR)
	BOOT.DBF (database)
	HMCOMP.DBF (database)
	N() (FP function)
	RESETWT() (function in HMCOMP)
	SELECT() ... (FP function)
	SELWT() (funct in HMCOMP)
	HMTAB.DBF (database)
	SELECT() ... (FP)
	RAND() (FP)
	IIF() ... (FP)
	EMPTY() ... (FP)
	DELFILE() (HMCOMP.SPR)
	CALCULATE() ... (HMCOMP.SPR)
	STD() (FP function)
	AVG() (FP function)
	ALLTRIM() ... (FP function)
	STR() ... (FP function)
	DELFILE() ... (HMCOMP.SPR)
DP()	... (function in DP.PRG)
SPACE()	(FP function)
BFACT()	(HMCOMP.SPR)
	BOOT.DBF (database)
	HMCOMP.DBF (database)
	PARAMETER() ... (FP function)
	SELECT() ... (FP function)
	N() ... (FP function)
	RESETWT() ... (HMCOMP.SPR)
	CALCULATE() ... (HMCOMP.SPR)
	STD() ... (FP function)
	AVG() ... (FP function)
	ALLTRIM() ... (FP function)
	STR() ... (FP function)
	DELFILE() ... (HMCOMP.SPR)
BWSTEP()	(HMCOMP.SPR)
	BOOT.DBF (database)
	HMCOMP.DBF (database)
	PARAMETER() ... (FP function)
	SELECT() ... (FP function)
	N() ... (FP function)
	RESETWT() ... (HMCOMP.SPR)



```

CALCULATE() ... (HMCOMP.SPR)
STD() ... (FP function)
AVG() ... (FP function)
ALLTRIM() ... (FP function)
STR() ... (FP function)
DELFILE() ... (HMCOMP.SPR)
BPHASE() (HMCOMP.SPR)
  BOOT.DBF (database)
  HMCOMP.DBF (database)
  PARAMETER() ... (FP function)
  SELECT() ... (FP function)
  N() ... (FP function)
  RESETWT() ... (HMCOMP.SPR)
  CALCULATE() (HMCOMP.SPR)
  STD() ... (FP function)
  AVG() ... (FP function)
  ALLTRIM() ... (FP function)
  STR() ... (FP function)
  DELFILE() (HMCOMP.SPR)
DISPLAY.PRG
  SET() ... (FP function)
  SELECT() ... (FP function)
  WEXIST() ... (FP function)
  INT() ... (FP function)
  SROW() ... (FP function)
  SCOL() ... (FP function)
  CHR() ... (FP function)
  EMPTY() ... (FP function)
  STR() ... (FP function)
  ALLEN() ... (FP function)
  DP() ... (function in DP.PRG)
  OCCURS() (FP function)
  WVISIBLE() ... (FP function)
  Q8X0JCQJY() (DISPLAY.PRG)
    W PRINT.SPR
      PARAMETER()
      SET()
      WEXIST()
      INT()
      SROW()
      SCOL()... (FP function)
      WVISIBLE().. (FP)
      Q9BOS33H5()
      Q9BOS33OA()
        PRINTFILE()
          ALLTRIM()
          EMPTY()
      Q9BOS33W4()
        OBJNUM()
  POPUPHIDE() (function in HMSC.PRG)
  HMBROWSE (procedure in HMCOMP.SPR)
    SELECT() ... (FP function)
    ALLTRIM() ... (FP function)
    UPPER() ... (FP function)
  DELFILE() ... (HMCOMP.SPR)
  CLOSEFILE (procedure in HMSC.PRG)
    EMPTY() ... (FP function)
  LOCFILE() ... (FP function)
  _Q9COH6DDQ (procedure in HMENU.MPR)

```

- BACKUP() (function in BACKUP.PRG)
- Q9COH6DEU (procedure in HMENU.MPR)
- ERRMSG() ... (function in ERRMSG.PRG)
- HMAT.SPR
  - HMAT.DBF (database)
  - SET() ... (FP function)
  - WEXIST() ... (FP function)
  - INT() ... (FP function)
  - SROW() ... (FP function)
  - SCOL() ... (FP function)
  - USED() ... (FP function)
  - TYPE() (FP function)
  - RECNO() ... (FP function)
  - WVISIBLE() ... (FP function)
  - Q9BORZLIV() (function in HMAT.SPR)
    - RECCOUNT() ... (FP function)
    - EOF() (FP function)
    - CHR() ... (FP function)
    - BOF() (FP function)
  - Q9BORZMCG() (function in HMAT.SPR)
    - RECNO() ... (FP function)
    - SEEK() (FP function)
    - FOUND() ... (FP function)
    - ERRMSG() ... (function in ERRMSG.PRG)
    - OBJNUM() ... (FP function)
  - Q9BORZMQY() (function in HMAT.SPR)
    - MEMOEDIT() (function in MEMOEDIT.PRG)
  - Q9BORZMXN() (function in HMAT.SPR)
    - MEMOEDIT() ... (function in MEMOEDIT.PRG)
  - Q9BORZN46() (function in HMAT.SPR)
    - MEMOEDIT() ... (function in MEMOEDIT.PRG)
  - Q9BORZNAQ() (function in HMAT.SPR)
    - MEMOEDIT() ... (function in MEMOEDIT.PRG)
  - Q9BORZNHA() (function in HMAT.SPR)
    - MEMOEDIT() ... (function in MEMOEDIT.PRG)
  - Q9BORZNP1() (function in HMAT.SPR)
    - MEMOEDIT() ... (function in MEMOEDIT.PRG)
  - Q9BORZNVN() (function in HMAT.SPR)
    - MEMOEDIT() ... (function in MEMOEDIT.PRG)
  - Q9BORZO25() (function in HMAT.SPR)
    - MEMOEDIT() ... (function in MEMOEDIT.PRG)
  - Q9BORZO8P() (function in HMAT.SPR)
    - MEMOEDIT() ... (function in MEMOEDIT.PRG)
  - Q9BORZOF9() (function in HMAT.SPR)
    - MEMOEDIT() ... (function in MEMOEDIT.PRG)
  - Q9BORZOLQ() (function in HMAT.SPR)
    - CHR() ... (FP function)
    - NEWLINE() ... (function in HMREF.PRG)
    - IIF() ... (FP function)
    - EMPTY() ... (FP function)
    - UPPER() ... (FP function)
    - ALLTRIM() ... (FP function)
    - MEMOEDIT() ... (function in MEMOEDIT.PRG)
  - Q9BORZP10() (function in HMAT.SPR)
    - MEMOEDIT() ... (function in MEMOEDIT.PRG)
  - Q9BORZPAH() (function in HMAT.SPR)
    - CHANGE (procedure in HMAT.SPR)
      - HMTAB.DBF (database)
      - SET() ... (FP function)

- TRIM() (FP function)
- USED() ... (FP function)
- UPPER() ... (FP function)
- ALLTRIM() ... (FP function)

#### HMLC.SPR

- HMLC.DBF (database)
- SET() ... (FP function)
- WEXIST() ... (FP function)
- INT() ... (FP function)
- SROW() ... (FP function)
- SCOL() ... (FP function)
- TYPE() ... (FP function)
- RECNO() ... (FP function)
- WVISIBLE() ... (FP function)
- Q9B0RZZSH() (function in HMLC.SPR)
  - RECCOUNT() ... (FP function)
  - EOF() ... (FP function)
  - CHR() ... (FP function)
  - BOF() ... (FP function)
- Q9B0S00F6() (function in HMLC.SPR)
  - RECNO() ... (FP function)
  - FOUND() ... (FP function)
  - ERRMSG() ... (function in ERRMSG.PRG)
- Q9B0S00OG() (function in HMLC.SPR)
- CHANGE... (procedure in HMLC.SPR)

#### HMWP.SPR

- HMWP.DBF (database)
- SET() ... (FP function)
- WEXIST() ... (FP function)
- INT() ... (FP function)
- SROW() ... (FP function)
- SCOL() ... (FP function)
- USED() ... (FP function)
- TYPE() ... (FP function)
- RECNO() ... (FP function)
- WVISIBLE() ... (FP function)
- Q9B0S091V() (function in HMWP.SPR)
  - RECCOUNT() ... (FP function)
  - EOF() ... (FP function)
  - CHR() ... (FP function)
  - BOF() ... (FP function)
- Q9B0S09L5() (function in HMWP.SPR)
  - RECNO() ... (FP function)
  - FOUND() ... (FP function)
  - ERRMSG() ... (function in ERRMSG.PRG)
  - OBJNUM() ... (FP function)
- Q9B0S09UL() (function in HMWP.SPR)
  - EMPTY() ... (FP function)
  - ERRMSG() ... (function in ERRMSG.PRG)
  - OBJNUM() ... (FP function)

#### HMET.SPR

- HMET.DBF (database)
- SET() ... (FP function)
- WEXIST() ... (FP function)
- INT() ... (FP function)
- SROW() ... (FP function)
- SCOL() ... (FP function)
- TYPE() ... (FP function)
- RECNO() ... (FP function)

- WVISIBLE() ... (FP function)
- Q9BOS0IUQ() (function in HMET.SPR)
  - RECCOUNT() ... (FP function)
  - EOF() ... (FP function)
  - CHR() ... (FP function)
  - BOF() ... (FP function)
- Q9BOS0JCN() (function in HMET.SPR)
  - CHANGE... (procedure in HMET.SPR)
- Q9BOS0JQ6() (function in HMET.SPR)
  - FOUND() ... (FP function)
  - ERRMSG() ... (function in ERRMSG.PRG)

#### HMCF.SPR

- HMCF.DBF (database)
- SET() ... (FP function)
- WEXIST() ... (FP function)
- INT() ... (FP function)
- SROW() ... (FP function)
- SCOL() ... (FP function)
- TYPE() ... (FP function)
- RECNO() ... (FP function)
- WVISIBLE() ... (FP function)
- Q9BOS0SSR() (function in HMCF.SPR)
  - RECCOUNT() ... (FP function)
  - EOF() ... (FP function)
  - CHR() ... (FP function)
  - BOF() ... (FP function)
- Q9BOS0TBW() (function in HMCF.SPR)
  - RECNO() ... (FP function)
  - FOUND() ... (FP function)
  - ERRMSG() ... (function in ERRMSG.PRG)
- Q9BOS0TKZ() (function in HMCF.SPR)
  - EMPTY() ... (FP function)
  - ERRMSG() ... (function in ERRMSG.PRG)
  - OBJNUM() ... (FP function)

#### HMCFE.SPR

- SET() ... (FP function)
- WEXIST() ... (FP function)
- INT() ... (FP function)
- SROW() ... (FP function)
- SCOL() ... (FP function)
- USED() ... (FP function)
- LOCFILE() ... (FP function)
- GET\_HMCF() (function in HMCFE.SPR)
  - HMCF.DBF (database)
  - SELECT() ... (FP function)
- WVISIBLE() ... (FP function)
- Q9BOS12JC() (function in HMCFE.SPR)
  - RECCOUNT() ... (FP function)
  - EOF() ... (FP function)
  - CHR() ... (FP function)
  - BOF() ... (FP function)
  - GET\_HMCF() ... (function in HMCFE.SPR)
- Q9BOS137V() (function in HMCFE.SPR)
- Q9BOS13DD() (function in HMCFE.SPR)
  - REL (procedure in HMCFE.SPR)
    - HMCFEI.DBF (database)
    - HMCFE.DBF (database)
    - HMCF.DBF (database)
- PARAMETER() ... (FP function)

```

      SELECT() ... (FP function)
      USED() ... (FP function)
      FOUND() ... (FP function)
      RECNO() ... (FP function)
      GET_HMCF() ... (function in HMCFE.SPR)
      Q9BOS13NF() (function in HMCFE.SPR)
      Q9BOS13SY() (function in HMCFE.SPR)
      REL...      (procedure in HMCFE.SPR)
      FOUND() ... (FP function)
      RECNO() ... (FP function)
      GET_HMCF() ... (function in HMCFE.SPR)
      Q9BOS1440() (function in HMCFE.SPR)
      Q9BOS14A5() (function in HMCFE.SPR)
      EMPTY() ... (FP function)
      ALLTRIM() ... (FP function)
      IIF() ... (FP function)
      UPPER() ... (FP function)
      GET_CF      (procedure in HMCFE.SPR)
      HMC.F.DBF (database)
      SELECT() ... (FP function)
      EMPTY() ... (FP function)
      CHOOSER() (function in CHOOSER.PRG)
      ERRMSG() ... (function in ERRMSG.PRG)
      GET_CFID    (procedure in HMCFE.SPR)
      HMC.F.DBF (database)
      PARAMETER() ... (FP function)
      SELECT() ... (FP function)
      Q9BOS14M9() (function in HMCFE.SPR)
      GET_HMCF() ... (function in HMCFE.SPR)
HMCFEI.SPR
      SET() ... (FP function)
      WEXIST() ... (FP function)
      INT() ... (FP function)
      SROW() ... (FP function)
      SCOL() ... (FP function)
      USED() ... (FP function)
      LOCFE() ... (FP function)
      GET_HMCF() ... (function in HMCFE.SPR)
      GET_HMCFE() (function in HMCFEI.SPR)
      HMC.FE.DBF (database)
      SELECT() ... (FP function)
      WVISIBLE() ... (FP function)
      Q9BOS1EP7() (function in HMCFEI.SPR)
      RECCOUNT() ... (FP function)
      EOF() ... (FP function)
      CHR() ... (FP function)
      BOF() ... (FP function)
      BROWSEITEM (procedure in HMCFEI.SPR)
      HMC.FE.DBF (database)
      HMC.FE.DBF (database)
      HMC.F.DBF (database)
      PARAMETER() ... (FP function)
      SELECT() ... (FP function)
      USED() ... (FP function)
      EMPTY() ... (FP function)
      FOUND() ... (FP function)
      RECNO() ... (FP function)
      GET_HMCF() ... (function in HMCFE.SPR)
      GET_HMCFE() ... (function in HMCFEI.SPR)

```

- Q9BOS1FQZ() (function in HMCFEI.SPR)
- Q9BOS1FWU() (function in HMCFEI.SPR)
- REL... (procedure in HMCFE.SPR)
- SELECT() ... (FP function)
- FOUND() ... (FP function)
- RECNO() ... (FP function)
- GET\_HMCF() ... (function in HMCFE.SPR)
- GET\_HMCFE() ... (function in HMCFEI.SPR)
- Q9BOS1G9W() (function in HMCFEI.SPR)
- Q9BOS1GFE() (function in HMCFEI.SPR)
- REL... (procedure in HMCFE.SPR)
- SELECT() ... (FP function)
- FOUND() ... (FP function)
- RECNO() ... (FP function)
- GET\_HMCF() ... (function in HMCFE.SPR)
- GET\_HMCFE() ... (function in HMCFEI.SPR)
- Q9BOS1GTD() (function in HMCFEI.SPR)
- Q9BOS1GZW() (function in HMCFEI.SPR)
- EMPTY() ... (FP function)
- ALLTRIM() ... (FP function)
- IIF() ... (FP function)
- UPPER() ... (FP function)
- GET\_CF... (procedure in HMCFE.SPR)
- GET\_CFD... (procedure in HMCFE.SPR)
- Q9BOS1HAP() (function in HMCFEI.SPR)
- Q9BOS1HGO() (function in HMCFEI.SPR)
- EMPTY() ... (FP function)
- ALLTRIM() ... (FP function)
- IIF() ... (FP function)
- UPPER() ... (FP function)
- GET\_CFE (procedure in HMCFEI.SPR)
  - HMCFE.DBF (database)
  - SELECT() ... (FP function)
  - EMPTY() ... (FP function)
  - CHOOSER() ... (function in CHOOSER.PRG)
  - ERRMSG() ... (function in ERRMSG.PRG)
- GET\_CFEID (procedure in HMCFEI.SPR)
  - HMCFE.DBF (database)
  - PARAMETER() ... (FP function)
  - SELECT() ... (FP function)
- Q9BOS1HTG() (function in HMCFEI.SPR)
- CHANGE... (procedure in HMTAB.SPR)
- GET\_HMCF() ... (function in HMCFE.SPR)
- GET\_HMCFE() ... (function in HMCFEI.SPR)
- HMTAB.SPR
  - HUNIT.DBF (database)
  - HMTAB.DBF (database)
  - SET() ... (FP function)
  - WEXIST() ... (FP function)
  - INT() ... (FP function)
  - SROW() ... (FP function)
  - SCOL() ... (FP function)
  - HMLU() ... (function in HMLU.PRG)
  - EMPTY() ... (FP function)
  - ERRMSG() ... (function in ERRMSG.PRG)
  - USED() ... (FP function)
  - GET\_HMID() (function in HMTAB.SPR)
    - HMTAB.DBF (database)
    - SELECT() ... (FP function)

```

—RECCOUNT() ... (FP function)
—IIF() ... (FP function)
—GET_HMLCN() (function in HMTAB.SPR)
  —HMLC.DBF (database)
  —SELECT() ... (FP function)
—GET_HMWP() (function in HMTAB.SPR)
  —HMWP.DBF (database)
  —SELECT() ... (FP function)
—GET_HMETN() (function in HMTAB.SPR)
  —HMET.DBF (database)
  —SELECT() ... (FP function)
—GET_CFAR() (function in HMTAB.SPR)
  —HMC.F.DBF (database)
  —SELECT() ... (FP function)
—GET_CFAR1() (function in HMTAB.SPR)
  —HMC.FE.DBF (database)
  —SELECT() ... (FP function)
—GET_EI() (function in HMTAB.SPR)
  —HMC.FEI.DBF (database)
  —SELECT() ... (FP function)
—WVISIBLE() ... (FP function)
—Q9B0S1VJD() (function in HMTAB.SPR)
  —RECCOUNT() ... (FP function)
  —SPACE() ... (FP function)
  —EMPTY() ... (FP function)
  —EOF() ... (FP function)
  —CHR() ... (FP function)
  —BOF() ... (FP function)
  —GET_HMLCN() ... (function in HMTAB.SPR)
  —GET_HMWP() ... (function in HMTAB.SPR)
  —GET_HMETN() ... (function in HMTAB.SPR)
  —GET_CFAR() ... (function in HMTAB.SPR)
  —GET_CFAR1() ... (function in HMTAB.SPR)
  —GET_EI() ... (function in HMTAB.SPR)
—Q9B0S1X8B() (function in HMTAB.SPR)
—Q9B0S1X10() (function in HMTAB.SPR)
  —HMLC.DBF (database)
  —EMPTY() ... (FP function)
  —ALLTRIM() ... (FP function)
  —IIF() ... (FP function)
  —UPPER() ... (FP function)
  —GET_HMLC() (function in HMTAB.SPR)
    —HMLC.DBF (database)
    —SELECT() ... (FP function)
    —EMPTY() ... (FP function)
    —CHOOSE() ... (function in CHOOSE.PRG)
    —ERRMSG() ... (function in ERRMSG.PRG)
    —ASCAN() ... (FP function)
    —STR() ... (FP function)
—Q9B0S1XTS() (function in HMTAB.SPR)
—Q9B0S1XZO() (function in HMTAB.SPR)
  —HMWP.DBF (database)
  —EMPTY() ... (FP function)
  —ALLTRIM() ... (FP function)
  —IIF() ... (FP function)
  —UPPER() ... (FP function)
  —GET_HMWP() (function in HMTAB.SPR)
    —HMWP.DBF (database)
    —SELECT() ... (FP function)

```

```

EMPTY() ... (FP function)
CHOOSE() ... (function in CHOOSE.PRG)
ERRMSG() ... (function in ERRMSG.PRG)
UPPER() ... (FP function)
ALLTRIM() ... (FP function)
ASCAN() ... (FP function)
STR() ... (FP function)
SPACE() ... (FP function)
Q9B0S1YBX() (function in HMTAB.SPR)
Q9B0S1YHS() (function in HMTAB.SPR)
HMET.DBF (database)
EMPTY() ... (FP function)
ALLTRIM() ... (FP function)
IIF() ... (FP function)
UPPER() ... (FP function)
GET_HMET() (function in HMTAB.SPR)
HMET.DBF (database)
SELECT() ... (FP function)
EMPTY() ... (FP function)
CHOOSE() ... (function in CHOOSE.PRG)
ERRMSG() ... (function in ERRMSG.PRG)
Q9B0S1YUM() (function in HMTAB.SPR)
Q9B0S1Z0J() (function in HMTAB.SPR)
HMC.F.DBF (database)
EMPTY() ... (FP function)
ALLTRIM() ... (FP function)
IIF() ... (FP function)
UPPER() ... (FP function)
GET CF... (procedure in HMC.F.SPR)
Q9B0S1ZDF() (function in HMTAB.SPR)
REL... (procedure in HMC.F.SPR)
EMPTY() ... (FP function)
GET_HMCFID() (function in HMTAB.SPR)
HMC.F.DBF (database)
SELECT() ... (FP function)
GET_EID() (function in HMTAB.SPR)
HMC.F.DBF (database)
SELECT() ... (FP function)
GET_EIID() (function in HMTAB.SPR)
HMC.FEI.DBF (database)
SELECT() ... (FP function)
Q9B0S1ZXN() (function in HMTAB.SPR)
CHANGE... (procedure in HMTAB.SPR)
GET_HMLCN() ... (function in HMTAB.SPR)
GET_HMWPN() ... (function in HMTAB.SPR)
GET_HMETN() ... (function in HMTAB.SPR)
GET_CFAR() ... (function in HMTAB.SPR)
GET_CFAR1() ... (function in HMTAB.SPR)
GET_EI() ... (function in HMTAB.SPR)
QUIT() (function in HMINIT.PRG)
MYHANDLER() (function in HMINIT.PRG)
TYPE() ... (FP function)

```



Section VI. Procedure and Function Summary. There are 25 files containing procedures in the HMLCCM system: HMINIT.PRG, HMENU.MPR, HMSC.PRG, &OLDPROC, HMREF.PRG, HMAT.SPR, HMLC.SPR, HMWP.SPR, HMET.SPR, HMC.F.SPR, HMC.FE.SPR, HMC.FEI.SPR, HMTAB.SPR, HMCOMP.SPR, DISPLAY.PRG, W.PRINT.SPR, BACKUP.PRG, HMLU.PRG, MEMOWIN.PRG, HMSCEN.PRG, HMSTEP.PRG, MEMOEDIT.PRG, CHOOSER.PRG, BSELECT.PRG, and BWFACT.PRG

1. HMINIT.PRG -- Last updated: 09/09/92 at 8:49

Contains: MYHANDLER() (Params: none)  
 Called by: HMINIT.PRG  
 Calls: TYPE() (FP function)  
 Contains: \_QUIT() (Params: none)  
 Called by: HMENU.MPR

2. HMENU.MPR -- Last updated: 09/24/92 at 8:00

Contains: \_Q9COH6DDQ (Params: none)  
 Called by: HMENU.MPR  
 Calls: BACKUP() (function in BACKUP.PRG)  
 Contains: \_Q9COH6DEU (Params: none)  
 Called by: HMENU.MPR  
 Calls: ERRMSG() (function in ERRMSG.PRG)

3. HMSC.PRG -- Last updated: 08/27/92 at 9:35

Contains: CLOSEFILE (Params: none)  
 Called by: HMSC.PRG  
 Calls: EMPTY() (FP function)  
 Contains: DEL\_HMSC() (Params: ID)  
 Called by: HMSC.PRG  
 Calls: OPENFILE() (function in HMSC.PRG)  
 Contains: FILEFIND() (Params: MFILENAME)  
 Called by: OPENFILE() (function in HMSC.PRG)  
 Calls: PARAMETER() (FP function)  
 Calls: FILE() (FP function)  
 Calls: DP() (function in DP.PRG)  
 Calls: LOCFILE() (FP function)  
 Contains: OPENFILE() (Params: FILE)  
 Called by: HMREF.PRG  
 Called by: DEL\_HMSC() (function in HMSC.PRG)  
 Called by: OPEN (procedure in HMCOMP.SPR)  
 Called by: HMSTEP.PRG  
 Calls: FILEFIND() (function in HMSC.PRG)  
 Calls: USED() (FP function)  
 Contains: POPUPSHOW() (Params: ERRSTR)  
 Called by: \_Q9BOS2HYD() (function in HMCOMP.SPR)  
 Calls: WEXIST() (FP function)  
 Calls: INT() (FP function)  
 Calls: SROW() (FP function)  
 Calls: SCOL() (FP function)  
 Calls: WVISIBLE() (FP function)  
 Contains: POPUPHIDE() (Params: W)  
 Called by: \_Q9BOS2HYD() (function in HMCOMP.SPR)

4. &OLDPROC --Temporary Variable

5. HMREF.PRG -- Last updated: 09/23/92 at 12:33

Contains: HMVIEW() (Params: NAME)

---

```

Called by: HMREF.PRG
Calls: PARAMETER()      (FP function)
Calls: FOUND()          (FP function)
Calls: CHR()            (FP function)
Calls: NEWLINE()        (function in HMREF.PRG)
Calls: IIF()            (FP function)
Calls: EMPTY()          (FP function)
Calls: UPPER()          (FP function)
Calls: ALLTRIM()        (FP function)
Calls: MEMOWIN()        (function in MEMOWIN.PRG)
Contains: NEWLINE()     (Params: TEXT)
Called by: HMVIEW()     (function in HMREF.PRG)
Called by: Q9BORZOLQ()  (function in HMAT.SPR)
Calls: CHR()            (FP function)
Calls: EMPTY()          (FP function)

```

6. HMAT.SPR -- Last updated: 09/23/92 at 13:03

```

Contains: CHANGE        (Params: none)
Called by: Q9BORZPAH()  (function in HMAT.SPR)
Called by: Q9BOS00OG()  (function in HMLC.SPR)
Called by: Q9BOS0JCN()  (function in HMET.SPR)
Called by: Q9BOS1HTG()  (function in HMCFEI.SPR)
Called by: Q9BOS1ZXN()  (function in HMTAB.SPR)
Called by: Q8ROY3L13()  (function in HMSTEP.PRG)
Calls: SET()            (FP function)
Calls: TRIM()           (FP function)
Calls: USED()           (FP function)
Calls: UPPER()          (FP function)
Calls: ALLTRIM()        (FP function)
Contains: NEWLINE()     (Params: TEXT)
Called by: HMVIEW()     (function in HMREF.PRG)
Called by: Q9BORZOLQ()  (function in HMAT.SPR)
Calls: CHR()            (FP function)
Calls: EMPTY()          (FP function)
Contains: Q9BORZLIV()   (Params: none)
Called By: HMAT.SPR
Calls: RECCOUNT()       (FP function)
Calls: EOF()            (FP function)
Calls: CHR()            (FP function)
Calls: BOF()            (FP function)
Contains: Q9BORZMCG()   (Params: none)
Called By: HMAT.SPR
Calls: RECNO()          (FP function)
Calls: SEEK()           (FP function)
Calls: FOUND()          (FP function)
Calls: ERRMSG()         (function in ERRMSG.PRG)
Calls: OBJNUM()         (FP function)
Contains: Q9BORZMQY()   (Params: none)
Called By: HMAT.SPR
Calls: MEMOEDIT()       (function MEMOEDIT.PRG)
Contains: Q9BORZMXN()   (Params: none)
Called By: HMAT.SPR
Calls: MEMOEDIT()       (function MEMOEDIT.PRG)
Contains: Q9BORZN46()   (Params: none)
Called By: HMAT.SPR
Calls: MEMOEDIT()       (MEMOEDIT.PRG)
Contains: Q9BORZNAQ()   (Params: none)
Called By: HMAT.SPR

```

---

Calls: MEMOEDIT()	(MEMOEDIT.PRG)
Contains: Q9BORZNHA()	(Params: none)
Called By: HMAT.SPR	
Calls: MEMOEDIT()	(MEMOEDIT.PRG)
Contains: Q9BORZNP1()	(Params: none)
Called By: HMAT.SPR	
Calls: MEMOEDIT()	(MEMOEDIT.PRG)
Contains: Q9BORZNVN()	(Params: none)
Called By: HMAT.SPR	
Calls: MEMOEDIT()	(MEMOEDIT.PRG)
Contains: Q9BORZO25()	(Params: none)
Called By: HMAT.SPR	
Calls: MEMOEDIT()	(MEMOEDIT.PRG)
Contains: Q9BORZO8P()	(Params: none)
Called By: HMAT.SPR	
Calls: MEMOEDIT()	(MEMOEDIT.PRG)
Contains: Q9BORZOF9()	(Params: none)
Called By: HMAT.SPR	
Calls: MEMOEDIT()	(MEMOEDIT.PRG)
Contains: Q9BORZOLQ()	(Params: none)
Called By: HMAT.SPR	
Calls: CHR()	(FP function)
Calls: NEWLINE()	(function in HMREF.PRG)
Calls: IIF()	(FP function)
Calls: EMPTY()	(FP function)
Calls: UPPER()	(FP function)
Calls: ALLTRIM()	(FP function)
Calls: MEMOEDIT()	(MEMOEDIT.PRG)
Contains: Q9BORZP10()	(Params: none)
Called By: HMAT.SPR	
Calls: MEMOEDIT()	(MEMOEDIT.PRG)
Contains: Q9BORZPAH()	(Params: none)
Called By: HMAT.SPR	
Calls: CHANGE	(procedure in HMAT.SPR)

---

7. HMLC.SPR -- Last updated: 09/23/92 at 13:03

Contains: CHANGE (Params: none)  
Called by: Q9BORZPAH() (function in HMAT.SPR)  
Called by: Q9BOS00OG() (function in HMLC.SPR)  
Called by: Q9BOS0JCN() (function in HMET.SPR)  
Called by: Q9BOS1HTG() (function in HMCFEI.SPR)  
Called by: Q9BOS1ZXN() (function in HMTAB.SPR)  
Called by: Q8ROY3L13() (function in HMSTEP.PRG)  
Calls: SET() (FP function)  
Calls: TRIM() (FP function)  
Calls: USED() (FP function)  
Calls: UPPER() (FP function)  
Calls: ALLTRIM() (FP function)  
Contains: Q9BORZZSH() (Params: none)  
Called By: HMLC.SPR  
Calls: RECCOUNT() (FP function)  
Calls: EOF() (FP function)  
Calls: CHR() (FP function)  
Calls: BOF() (FP function)  
Contains: Q9BOS00F6() (Params: none)  
Called By: HMLC.SPR  
Calls: RECNO() (FP function)  
Calls: FOUND() (FP function)  
Calls: ERRMSG() (function in ERRMSG.PRG)  
Contains: Q9BOS00OG() (Params: none)  
Called By: HMLC.SPR  
Calls: CHANGE (procedure in HMAT.SPR)

8. HMWP.SPR -- Last updated: 09/23/92 at 13:04

Contains: CHANGE (Params: none)  
Called by: Q9BORZPAH() (function in HMAT.SPR)  
Called by: Q9BOS00OG() (function in HMLC.SPR)  
Called by: Q9BOS0JCN() (function in HMET.SPR)  
Called by: Q9BOS1HTG() (function in HMCFEI.SPR)  
Called by: Q9BOS1ZXN() (function in HMTAB.SPR)  
Called by: Q8ROY3L13() (function in HMSTEP.PRG)  
Calls: SET() (FP function)  
Calls: TRIM() (FP function)  
Calls: USED() (FP function)  
Calls: UPPER() (FP function)  
Calls: ALLTRIM() (FP function)  
Contains: Q9BOS091V() (Params: none)  
Called By: HMWP.SPR  
Calls: RECCOUNT() (FP function)  
Calls: EOF() (FP function)  
Calls: CHR() (FP function)  
Calls: BOF() (FP function)  
Contains: Q9BOS09L5() (Params: none)  
Called By: HMWP.SPR  
Calls: RECNO() (FP function)  
Calls: FOUND() (FP function)  
Calls: ERRMSG() (function in ERRMSG.PRG)  
Calls: OBJNUM() (FP function)  
Contains: Q9BOS09UL() (Params: none)  
Called By: HMWP.SPR  
Calls: EMPTY() (FP function)

---

Calls: ERRMSG() (function in ERRMSG.PRG)  
Calls: OBJNUM() (FP function)

9. HMET.SPR -- Last updated: 09/23/92 at 13:04

Contains: CHANGE (Params: none)  
Called by: \_Q9BORZPAH() (function in HMAT.SPR)  
Called by: \_Q9BOS00OG() (function in HMLC.SPR)  
Called by: \_Q9BOS0JCN() (function in HMET.SPR)  
Called by: \_Q9BOS1HTG() (function in HMCFEI.SPR)  
Called by: \_Q9BOS1ZXN() (function in HMTAB.SPR)  
Called by: \_Q8ROY3L13() (function in HMSTEP.PRG)  
Calls: SET() (FP function)  
Calls: TRIM() (FP function)  
Calls: USED() (FP function)  
Calls: UPPER() (FP function)  
Calls: ALLTRIM() (FP function)  
Contains: \_Q9BOS0IUQ() (Params: none)  
Called by: HMET.SPR  
Calls: RECCOUNT() (FP function)  
Calls: EOF() (FP function)  
Calls: CHR() (FP function)  
Calls: BOF() (FP function)  
Contains: \_Q9BOS0JCN() (Params: none)  
Called by: HMET.SPR  
Calls: CHANGE (procedure in HMAT.SPR)  
Contains: \_Q9BOS0JQ6() (Params: none)  
Called by: HMET.SPR  
Calls: FOUND() (FP function)  
Calls: ERRMSG() (function in ERRMSG.PRG)

10. HMC.F.SPR -- Last updated: 09/23/92 at 13:04

Contains: CHANGE (Params: none)  
Called by: \_Q9BORZPAH() (function in HMAT.SPR)  
Called by: \_Q9BOS00OG() (function in HMLC.SPR)  
Called by: \_Q9BOS0JCN() (function in HMET.SPR)  
Called by: \_Q9BOS1HTG() (function in HMCFEI.SPR)  
Called by: \_Q9BOS1ZXN() (function in HMTAB.SPR)  
Called by: \_Q8ROY3L13() (function in HMSTEP.PRG)  
Calls: SET() (FP function)  
Calls: TRIM() (FP function)  
Calls: USED() (FP function)  
Calls: UPPER() (FP function)  
Calls: ALLTRIM() (FP function)  
Contains: \_Q9BOS0SSR() (Params: none)  
Called by: HMC.F.SPR  
Calls: RECCOUNT() (FP function)  
Calls: EOF() (FP function)  
Calls: CHR() (FP function)  
Calls: BOF() (FP function)  
Contains: \_Q9BOS0TSW() (Params: none)  
Called by: HMC.F.SPR  
Calls: RECNO() (FP function)  
Calls: FOUND() (FP function)  
Calls: ERRMSG() (function in ERRMSG.PRG)  
Contains: \_Q9BOS0TKZ() (Params: none)  
Called by: HMC.F.SPR  
Calls: EMPTY() (FP function)

Calls: ERRMSG() (function in ERRMSG.PRG)  
 Calls: OBJNUM() (FP function)

11. HMCFE.SPR -- Last updated: 09/23/92 at 13:04

Contains: ESCPRESSED (Params: none)  
 Contains: CHANGE (Params: none)  
   Called by: \_Q9BORZPAH() (function in HMAT.SPR)  
   Called by: \_Q9BOS00OG() (function in HMLC.SPR)  
   Called by: \_Q9BOS0JCN() (function in HMET.SPR)  
   Called by: \_Q9BOS1HTG() (function in HMCFEI.SPR)  
   Called by: \_Q9BOS1ZXN() (function in HMTAB.SPR)  
   Called by: \_Q8ROY3L13() (function in HMSTEP.PRG)  
   Calls: SET() (FP function)  
   Calls: TRIM() (FP function)  
   Calls: USED() (FP function)  
   Calls: UPPER() (FP function)  
   Calls: ALLTRIM() (FP function)  
 Contains: GET\_HMCF() (Params: M.HMCFID)  
   Called by: HMCFE.SPR  
   Called by: HMCFEI.SPR  
   Called by: \_Q9BOS12JC() (function in HMCFE.SPR)  
   Called by: \_Q9BOS13DD() (function in HMCFE.SPR)  
   Called by: \_Q9BOS13SY() (function in HMCFE.SPR)  
   Called by: \_Q9BOS14M9() (function in HMCFE.SPR)  
   Called by: \_Q9BOS1EP7() (function in HMCFEI.SPR)  
   Called by: \_Q9BOS1FWU() (function in HMCFEI.SPR)  
   Called by: \_Q9BOS1GFE() (function in HMCFEI.SPR)  
   Called by: \_Q9BOS1HTG() (function in HMCFEI.SPR)  
   Calls: SELECT() (FP function)  
 Contains: GET\_CF (Params: M.HMCF)  
   Called by: \_Q9BOS14A5() (function in HMCFE.SPR)  
   Called by: \_Q9BOS1GZW() (function in HMCFEI.SPR)  
   Called by: \_Q9BOS1Z0J() (function in HMTAB.SPR)  
   Calls: SELECT() (FP function)  
   Calls: EMPTY() (FP function)  
   Calls: CHOOSER() (function CHOOSER.PRG)  
   Calls: ERRMSG() (function in ERRMSG.PRG)  
 Contains: GET\_CFID (Params: M.ANSWR)  
   Called by: \_Q9BOS14A5() (function in HMCFE.SPR)  
   Called by: \_Q9BOS1GZW() (function in HMCFEI.SPR)  
   Calls: PARAMETER() (FP function)  
   Calls: SELECT() (FP function)  
 Contains: REL (Params: M.ID)  
   Called by: \_Q9BOS13DD() (function in HMCFE.SPR)  
   Called by: \_Q9BOS13SY() (function in HMCFE.SPR)  
   Called by: \_Q9BOS1FWU() (function in HMCFEI.SPR)  
   Called by: \_Q9BOS1GFE() (function in HMCFEI.SPR)  
   Called by: \_Q9BOS1ZDF() (function in HMTAB.SPR)  
   Calls: PARAMETER() (FP function)  
   Calls: SELECT() (FP function)  
   Calls: USED() (FP function)  
 Contains: \_Q9BOS12JC() (Params: none)  
   Called by: HMCFE.SPR  
   Calls: RECCOUNT() (FP function)  
   Calls: EOF() (FP function)  
   Calls: CHR() (FP function)  
   Calls: BOF() (FP function)  
   Calls: GET\_HMCF() (function in HMCFE.SPR)

---

Contains: Q9BOS137V()	(Params: none)
Called By: HMCFE.SPR	
Contains: Q9BOS13DD()	(Params: none)
Called By: HMCFE.SPR	
Calls: REL	(procedure in HMCFE.SPR)
Calls: FOUND()	(FP function)
Calls: RECNO()	(FP function)
Calls: GET_HMCF()	(function in HMCFE.SPR)
Contains: Q9BOS13NF()	(Params: none)
Called By: HMCFE.SPR	
Contains: Q9BOS13SY()	(Params: none)
Called By: HMCFE.SPR	
Calls: REL	(procedure in HMCFE.SPR)
Calls: FOUND()	(FP function)
Calls: RECNO()	(FP function)
Calls: GET_HMCF()	(function in HMCFE.SPR)
Contains: Q9BOS1440()	(Params: none)
Called By: HMCFE.SPR	
Contains: Q9BOS14A5()	(Params: none)
Called By: HMCFE.SPR	
Calls: EMPTY()	(FP function)
Calls: ALLTRIM()	(FP function)
Calls: IIF()	(FP function)
Calls: UPPER()	(FP function)
Calls: GET CF	(procedure in HMCFE.SPR)
Calls: GET CFID	(procedure in HMCFE.SPR)
Contains: Q9BOS14M9()	(Params: none)
Called By: HMCFE.SPR	
Calls: GET_HMCF()	(function in HMCFE.SPR)

12. HMCFEI.SPR -- Last updated: 09/23/92 at 13:05

Contains: ESCPRESSED	(Params: none)
Contains: CHANGE	(Params: none)
Called by: _Q9BORZPAH()	(function in HMAT.SPR)
Called by: _Q9BOS00OG()	(function in HMLC.SPR)
Called by: _Q9BOS0JCN()	(function in HMET.SPR)
Called by: _Q9BOS1HTG()	(function in HMCFEI.SPR)
Called by: _Q9BOS1ZXN()	(function in HMTAB.SPR)
Called by: _Q8ROY3L13()	(function in HMSTEP.PRG)
Calls: SET()	(FP function)
Calls: TRIM()	(FP function)
Calls: USED()	(FP function)
Calls: UPPER()	(FP function)
Calls: ALLTRIM()	(FP function)
Contains: GET_HMCF()	(Params: M.HMCFID)
Called by: HMCFE.SPR	
Called by: HMCFEI.SPR	
Called by: _Q9BOS12JC()	(function in HMCFE.SPR)
Called by: _Q9BOS13DD()	(function in HMCFE.SPR)
Called by: _Q9BOS13SY()	(function in HMCFE.SPR)
Called by: _Q9BOS14M9()	(function in HMCFE.SPR)
Called by: _Q9BOS1EP7()	(function in HMCFEI.SPR)
Called by: _Q9BOS1FWU()	(function in HMCFEI.SPR)
Called by: _Q9BOS1GFE()	(function in HMCFEI.SPR)
Called by: _Q9BOS1HTG()	(function in HMCFEI.SPR)
Calls: SELECT()	(FP function)
Contains: GET_CF	(Params: M.HMCF)
Called by: _Q9BOS14A5()	(function in HMCFE.SPR)
Called by: _Q9BOS1GZW()	(function in HMCFEI.SPR)
Called by: _Q9BOS1Z0J()	(function in HMTAB.SPR)
Calls: SELECT()	(FP function)
Calls: EMPTY()	(FP function)
Calls: CHOOSER()	(CHOOSER.PRG)
Calls: ERRMSG()	(function in ERRMSG.PRG)
Contains: GET_CFID	(Params: M.ANSWR)
Called by: _Q9BOS14A5()	(function in HMCFE.SPR)
Called by: _Q9BOS1GZW()	(function in HMCFEI.SPR)
Calls: PARAMETER()	(FP function)
Calls: SELECT()	(FP function)
Contains: GET_HMCFE()	(Params: M.HMCFEID)
Called by: HMCFEI.SPR	
Called by: _Q9BOS1EP7()	(function in HMCFEI.SPR)
Called by: _Q9BOS1FWU()	(function in HMCFEI.SPR)
Called by: _Q9BOS1GFE()	(function in HMCFEI.SPR)
Called by: _Q9BOS1HTG()	(function in HMCFEI.SPR)
Calls: SELECT()	(FP function)
Contains: GET_CFE	(Params: M.HMCFE)
Called by: _Q9BOS1HGO()	(function in HMCFEI.SPR)
Calls: SELECT()	(FP function)
Calls: EMPTY()	(FP function)
Calls: CHOOSER()	(CHOOSER.PRG)
Calls: ERRMSG()	(function in ERRMSG.PRG)
Contains: GET_CFEID	(Params: M.HMCFE)
Called by: _Q9BOS1HGO()	(function in HMCFEI.SPR)
Calls: PARAMETER()	(FP function)
Calls: SELECT()	(FP function)
Contains: REL	(Params: M.ID)
Called by: _Q9BOS13DD()	(function in HMCFE.SPR)



Called by: Q9BOS13SY()	(function in HMCFE.SPR)
Called by: Q9BOS1FWU()	(function in HMCFEI.SPR)
Called by: Q9BOS1GFE()	(function in HMCFEI.SPR)
Called by: Q9BOS1ZDF()	(function in HMTAB.SPR)
Calls: PARAMETER()	(FP function)
Calls: SELECT()	(FP function)
Calls: USED()	(FP function)
Contains: BRO SEITEM	(Params: M.ID)
Called by: Q9BOS1EP7()	(function in HMCFEI.SPR)
Calls: PARAMETER()	(FP function)
Calls: SELECT()	(FP function)
Calls: USED()	(FP function)
Calls: EMPTY()	(FP function)
Contains: Q9BOS1EP7()	(Params: none)
Called by: HMCFEI.SPR	
Calls: RECCOUNT()	(FP function)
Calls: EOF()	(FP function)
Calls: CHR()	(FP function)
Calls: BOF()	(FP function)
Calls: BROWSEITEM	(procedure in HMCFEI.SPR)
Calls: FOUND()	(FP function)
Calls: RECNO()	(FP function)
Calls: GET_HMCF()	(function in HMCFE.SPR)
Calls: GET_HMCFE()	(function in HMCFEI.SPR)
Contains: Q9BOS1FQZ()	(Params: none)
Called by: HMCFEI.SPR	
Contains: Q9BOS1FWU()	(Params: none)
Called by: HMCFEI.SPR	
Calls: REL	(procedure in HMCFE.SPR)
Calls: SELECT()	(FP function)
Calls: FOUND()	(FP function)
Calls: RECNO()	(FP function)
Calls: GET_HMCF()	(function in HMCFE.SPR)
Calls: GET_HMCFE()	(function in HMCFEI.SPR)
Contains: Q9BOS1G9W()	(Params: none)
Called by: HMCFEI.SPR	
Contains: Q9BOS1GFE()	(Params: none)
Called by: HMCFEI.SPR	
Calls: REL	(procedure in HMCFE.SPR)
Calls: SELECT()	(FP function)
Calls: FOUND()	(FP function)
Calls: RECNO()	(FP function)
Calls: GET_HMCF()	(function in HMCFE.SPR)
Calls: GET_HMCFE()	(function in HMCFEI.SPR)
Contains: Q9BOS1GTD()	(Params: none)
Called by: HMCFEI.SPR	
Contains: Q9BOS1GZW()	(Params: none)
Called by: HMCFEI.SPR	
Calls: EMPTY()	(FP function)
Calls: ALLTRIM()	(FP function)
Calls: IIF()	(FP function)
Calls: UPPER()	(FP function)
Calls: GET_CF	(procedure in HMCFE.SPR)
Calls: GET_CPID	(procedure in HMCFE.SPR)
Contains: Q9BOS1HAP()	(Params: none)
Called by: HMCFEI.SPR	
Contains: Q9BOS1HGO()	(Params: none)
Called by: HMCFEI.SPR	
Calls: EMPTY()	(FP function)

---

Calls: ALLTRIM()	(FP function)
Calls: IIF()	(FP function)
Calls: UPPER()	(FP function)
Calls: GET_CFE	(procedure in HMCFEI.SPR)
Calls: GET_CFEID	(procedure in HMCFEI.SPR)
Contains: Q9BOS1HTG()	(Params: none)
Called by: HMCFEI.SPR	
Calls: CHANGE	(procedure in HMAT.SPR)
Calls: GET_HMCF()	(function in HMCFE.SPR)
Calls: GET_HMCFE()	(function in HMCFEI.SPR)

13. HMTAB.SPR -- Last updated: 09/23/92 at 13:05

Contains: ESCPRESSED	(Params: none)
Contains: CHANGE	(Params: none)
Called by: Q9BORZPAH()	(function in HMAT.SPR)
Called by: Q9BOS00OG()	(function in HMLC.SPR)
Called by: Q9BOS0JCN()	(function in HMET.SPR)
Called by: Q9BOS1HTG()	(function in HMCFEI.SPR)
Called by: Q9BOS1ZXN()	(function in HMTAB.SPR)
Called by: Q8ROY3L13()	(function in HMSTEP.PRG)
Calls: SET()	(FP function)
Calls: TRIM()	(FP function)
Calls: USED()	(FP function)
Calls: UPPER()	(FP function)
Calls: ALLTRIM()	(FP function)
Contains: GET_HMNAME()	(Params: none)
Called by: Q7DOO1EI8()	(function in HMLU.PRG)
Calls: SELECT()	(FP function)
Calls: EMPTY()	(FP function)
Calls: CHOOSER()	(CHOOSER.PRG)
Calls: ERRMSG()	(function in ERRMSG.PRG)
Contains: GET_HMID()	(Params: M.HMNAME)
Called by: HMTAB.SPR	
Calls: SELECT()	(FP function)
Contains: GET_HMLC()	(Params: M.HMLC)
Called by: Q9BOS1XI0()	(function in HMTAB.SPR)
Called by: Q8ROY3JYQ()	(function in HMSTEP.PRG)
Calls: SELECT()	(FP function)
Calls: EMPTY()	(FP function)
Calls: CHOOSER()	(CHOOSER.PRG)
Calls: ERRMSG()	(function in ERRMSG.PRG)
Calls: ASCAN()	(FP function)
Calls: STR()	(FP function)
Contains: GET_HMLCN()	(Params: M.ID)
Called by: HMTAB.SPR	
Called by: Q9BOS1VJD()	(function in HMTAB.SPR)
Called by: Q9BOS1ZXN()	(function in HMTAB.SPR)
Called by: INITVAR	(procedure in HMSTEP.PRG)
Calls: SELECT()	(FP function)
Contains: GET_HMWP()	(Params: M.MATCH)
Called by: Q9BOS1XZO()	(function in HMTAB.SPR)
Called by: Q8ROY3KIH()	(function in HMSTEP.PRG)
Calls: SELECT()	(FP function)
Calls: EMPTY()	(FP function)
Calls: CHOOSER()	(CHOOSER.PRG)
Calls: ERRMSG()	(function in ERRMSG.PRG)
Calls: UPPER()	(FP function)
Calls: ALLTRIM()	(FP function)

Calls: ASCAN()	(FP function)
Calls: STR()	(FP function)
Contains: GET_HMWP()	(Params: M.ID)
Called by: HMTAB.SPR	
Called by: _Q9B0S1VJD()	(function in HMTAB.SPR)
Called by: _Q9B0S1ZXN()	(function in HMTAB.SPR)
Called by: INITVAR	(procedure in HMSTEP.PRG)
Calls: SELECT()	(FP function)
Contains: GET_HMET()	(Params: M.HMET)
Called by: _Q9B0S1YHS()	(function in HMTAB.SPR)
Calls: SELECT()	(FP function)
Calls: EMPTY()	(FP function)
Calls: CHOOSER()	(CHOOSER.PRG)
Calls: ERRMSG()	(function in ERRMSG.PRG)
Contains: GET_HMETID()	(Params: M.HMET)
Calls: SELECT()	(FP function)
Contains: GET_HMETN()	(Params: M.HMETID)
Called by: HMTAB.SPR	
Called by: _Q9B0S1VJD()	(function in HMTAB.SPR)
Called by: _Q9B0S1ZXN()	(function in HMTAB.SPR)
Calls: SELECT()	(FP function)
Contains: GET_HCFAR()	(Params: M.HMCFID)
Called by: HMTAB.SPR	
Called by: _Q9B0S1VJD()	(function in HMTAB.SPR)
Called by: _Q9B0S1ZXN()	(function in HMTAB.SPR)
Calls: SELECT()	(FP function)
Contains: GET_HCFAR1()	(Params: M.HMCFEID)
Called by: HMTAB.SPR	
Called by: _Q9B0S1VJD()	(function in HMTAB.SPR)
Called by: _Q9B0S1ZXN()	(function in HMTAB.SPR)
Calls: SELECT()	(FP function)
Contains: GET_HCF	(Params: M.HMCF)
Called by: _Q9B0S14A5()	(function in HMCFE.SPR)
Called by: _Q9B0S1GZW()	(function in HMCFEI.SPR)
Called by: _Q9B0S1Z0J()	(function in HMTAB.SPR)
Calls: SELECT()	(FP function)
Calls: EMPTY()	(FP function)
Calls: CHOOSER()	(CHOOSER.PRG)
Calls: ERRMSG()	(function in ERRMSG.PRG)
Contains: GET_HMCFID()	(Params: M.HMCF)
Called by: _Q9B0S1ZDF()	(function in HMTAB.SPR)
Calls: SELECT()	(FP function)
Contains: GET_HCF1()	(Params: M.HMCFE)
Calls: SELECT()	(FP function)
Calls: EMPTY()	(FP function)
Calls: CHOOSER()	(CHOOSER.PRG)
Calls: ERRMSG()	(function in ERRMSG.PRG)
Contains: GET_HEID()	(Params: M.HMCFE)
Called by: _Q9B0S1ZDF()	(function in HMTAB.SPR)
Calls: SELECT()	(FP function)
Contains: GET_HEI()	(Params: M.HMCFEIID)
Called by: HMTAB.SPR	
Called by: _Q9B0S1VJD()	(function in HMTAB.SPR)
Called by: _Q9B0S1ZXN()	(function in HMTAB.SPR)
Calls: SELECT()	(FP function)
Contains: GET_HEIID()	(Params: M.HMCFEII)
Called by: _Q9B0S1ZDF()	(function in HMTAB.SPR)
Calls: SELECT()	(FP function)
Contains: REL	(Params: M.ID)

Called by: Q9B0S13DD()	(function in HMCFE.SPR)
Called by: Q9B0S13SY()	(function in HMCFE.SPR)
Called by: Q9B0S1FWU()	(function in HMCFEI.SPR)
Called by: Q9B0S1GFE()	(function in HMCFEI.SPR)
Called by: Q9B0S1ZDF()	(function in HMTAB.SPR)
Calls: PARAMETER()	(FP function)
Calls: SELECT()	(FP function)
Calls: USED()	(FP function)
Contains: Q9B0S1VJD()	(Params: none)
Called By: HMTAB.SPR	
Calls: RECCOUNT()	(FP function)
Calls: SPACE()	(FP function)
Calls: EMPTY()	(FP function)
Calls: EOF()	(FP function)
Calls: CHR()	(FP function)
Calls: BOF()	(FP function)
Calls: GET_HMLCN()	(function in HMTAB.SPR)
Calls: GET_HMWPN()	(function in HMTAB.SPR)
Calls: GET_HMETN()	(function in HMTAB.SPR)
Calls: GET_CFAR()	(function in HMTAB.SPR)
Calls: GET_CFAR1()	(function in HMTAB.SPR)
Calls: GET_EI()	(function in HMTAB.SPR)
Contains: Q9B0S1X8B()	(Params: none)
Called By: HMTAB.SPR	
Contains: Q9B0S1XI0()	(Params: none)
Called By: HMTAB.SPR	
Calls: EMPTY()	(FP function)
Calls: ALLTRIM()	(FP function)
Calls: IIF()	(FP function)
Calls: UPPER()	(FP function)
Calls: GET_HMLC()	(function in HMTAB.SPR)
Contains: Q9B0S1XTS()	(Params: none)
Called By: HMTAB.SPR	
Contains: Q9B0S1XZO()	(Params: none)
Called By: HMTAB.SPR	
Calls: EMPTY()	(FP function)
Calls: ALLTRIM()	(FP function)
Calls: IIF()	(FP function)
Calls: UPPER()	(FP function)
Calls: GET_HMWP()	(function in HMTAB.SPR)
Calls: SPACE()	(FP function)
Contains: Q9B0S1YBX()	(Params: none)
Called By: HMTAB.SPR	
Contains: Q9B0S1YHS()	(Params: none)
Called By: HMTAB.SPR	
Calls: EMPTY()	(FP function)
Calls: ALLTRIM()	(FP function)
Calls: IIF()	(FP function)
Calls: UPPER()	(FP function)
Calls: GET_HMET()	(function in HMTAB.SPR)
Contains: Q9B0S1YUM()	(Params: none)
Called By: HMTAB.SPR	
Contains: Q9B0S1Z0J()	(Params: none)
Called By: HMTAB.SPR	
Calls: EMPTY()	(FP function)
Calls: ALLTRIM()	(FP function)
Calls: IIF()	(FP function)
Calls: UPPER()	(FP function)
Calls: GET_CF	(procedure in HMCFE.SPR)

Contains: Q9BOS1ZDF()	(Params: none)
Called By: HMTAB.SPR	
Calls: REL	(procedure in HMCFE.SPR)
Calls: EMPTY()	(FP function)
Calls: GET_HMCFID()	(function in HMTAB.SPR)
Calls: GET_EID()	(function in HMTAB.SPR)
Calls: GET_EIID()	(function in HMTAB.SPR)
Contains: Q9BOS1ZXN()	(Params: none)
Called By: HMTAB.SPR	
Calls: CHANGE	(procedure in HMTAB.SPR)
Calls: GET_HMLCN()	(function in HMTAB.SPR)
Calls: GET_HMWPIN()	(function in HMTAB.SPR)
Calls: GET_HMETN()	(function in HMTAB.SPR)
Calls: GET_C FAR()	(function in HMTAB.SPR)
Calls: GET_C FAR1()	(function in HMTAB.SPR)
Calls: GET_EI()	(function in HMTAB.SPR)

14. HMCMP.SPR -- Last updated: 09/23/92 at 13:05

Contains: OPEN	(Params: none)
Called by: HMCMP.SPR	
Calls: OPENFILE()	(function in HMSC.PRG)
Calls: CANCEL	(procedure in HMCMP.SPR)
Contains: DELFILE()	(Params: FILE)
Called by: HMCMP.SPR	
Called by: CALCULATE()	(function in HMCMP.SPR)
Called by: SELWT()	(function in HMCMP.SPR)
Called by: BSTEP()	(function in HMCMP.SPR)
Called by: BFACT()	(function in HMCMP.SPR)
Called by: BWSTEP()	(function in HMCMP.SPR)
Called by: BPHASE()	(function in HMCMP.SPR)
Calls: USED()	(FP function)
Contains: CANCEL	(Params: SUCCESS)
Called by: OPEN	(procedure in HMCMP.SPR)
Called by: CANCEL	(procedure in HMCMP.SPR)
Called by: HMSTEP.PRG	
Calls: ERRMSG()	(function in ERRMSG.PRG)
Calls: CANCEL	(procedure in HMCMP.SPR)
Contains: GET_HMARRAY	(Params: none)
Called by: HMCMP.SPR	
Calls: SELECT()	(FP function)
Calls: RECCOUNT()	(FP function)
Calls: STR()	(FP function)
Calls: RECNO()	(FP function)
Calls: GET TABLE	(procedure in HMCMP.SPR)
Calls: ALLTRIM()	(FP function)
Calls: ASCAN()	(FP function)
Calls: AINS()	(FP function)
Contains: GET TABLE	(Params: MATID,LCID,WPID)
Called by: GET_HMARRAY	(procedure in HMCMP.SPR)
Calls: SELECT()	(FP function)
Contains: COMPUTE	(Params: none)
Called by: Q9BOS2HYD()	(function in HMCMP.SPR)
Calls: SELECT()	(FP function)
Calls: RECCOUNT()	(FP function)
Calls: COMPUTSTEP()	(function in HMCMP.SPR)
Calls: ALLTRIM()	(FP function)
Calls: STR()	(FP function)
Calls: EMPTY()	(FP function)

---

Calls: ALLEN()	(FP function)
Calls: COMPUTFACT()	(function in HMCOMP.SPR)
Calls: COMPUTWFACT()	(function in HMCOMP.SPR)
Calls: COMPUTPHASE()	(function in HMCOMP.SPR)
Calls: CHR()	(FP function)
Calls: BSTEP()	(function in HMCOMP.SPR)
Calls: DP()	(function in DP.PRG)
Calls: SPACE()	(FP function)
Calls: BFACT()	(function in HMCOMP.SPR)
Calls: BWSTEP()	(function in HMCOMP.SPR)
Calls: BPHASE()	(function in HMCOMP.SPR)
Calls: DISPLAY.PRG	
Contains: HMBROWSE	(Params: none)
Called by: Q9B0S2HYD()	(function in HMCOMP.SPR)
Calls: SELECT()	(FP function)
Calls: ALLTRIM()	(FP function)
Calls: UPPER()	(FP function)
Contains: COMPUTSTEP()	(Params: MAT, LCPHASE, WP, SN, SD, SQ)
Called by: COMPUTE	(procedure in HMCOMP.SPR)
Calls: SELECT()	(FP function)
Calls: CALCULATE()	(function in HMCOMP.SPR)
Contains: COMPUTFACT()	(Params: M.FACTID)
Called by: COMPUTE	(procedure in HMCOMP.SPR)
Calls: PARAMETER()	(FP function)
Calls: SELECT()	(FP function)
Calls: CALCULATE()	(function in HMCOMP.SPR)
Contains: COMPUTWFACT()	(Params: M.FACTID)
Called by: COMPUTE	(procedure in HMCOMP.SPR)
Calls: PARAMETER()	(FP function)
Calls: SELECT()	(FP function)
Calls: IIF()	(FP function)
Calls: EMPTY()	(FP function)
Calls: ALLTRIM()	(FP function)
Calls: STR()	(FP function)
Calls: CALCULATE()	(function in HMCOMP.SPR)
Contains: COMPUTPHASE()	(Params: M.PHASEID)
Called by: COMPUTE	(procedure in HMCOMP.SPR)
Calls: PARAMETER()	(FP function)
Calls: SELECT()	(FP function)
Calls: CALCULATE()	(function in HMCOMP.SPR)
Contains: CALCULATE()	(Params: SN, SD, SQ)
Called by: COMPUTSTEP()	(function in HMCOMP.SPR)
Called by: COMPUTFACT()	(function in HMCOMP.SPR)
Called by: COMPUTWFACT()	(function in HMCOMP.SPR)
Called by: COMPUTPHASE()	(function in HMCOMP.SPR)
Called by: BSTEP()	(function in HMCOMP.SPR)
Called by: BFACT()	(function in HMCOMP.SPR)
Called by: BWSTEP()	(function in HMCOMP.SPR)
Called by: BPHASE()	(function in HMCOMP.SPR)
Calls: SELECT()	(FP function)
Calls: USED()	(FP function)
Calls: RECCOUNT()	(FP function)
Calls: EMPTY()	(FP function)
Calls: IIF()	(FP function)
Calls: RECNO()	(FP function)
Calls: DELFILE()	(function in HMCOMP.SPR)
Contains: RESETWT()	(Params: MAT, LCPHASE, WP)
Called by: BSTEP()	(function in HMCOMP.SPR)

---

Called by: BFACT()	(function in HMCOMP.SPR)
Called by: BWSTEP()	(function in HMCOMP.SPR)
Called by: BPHASE()	(function in HMCOMP.SPR)
Calls: SELECT()	(FP function)
Calls: SELWT()	(function in HMCOMP.SPR)
Contains: SELWT()	(Params: MAT, LCPHASE, WP, METID, CFID, CFEID)
Called by: RESETWT()	(function in HMCOMP.SPR)
Calls: SELECT()	(FP function)
Calls: RAND()	(FP function)
Calls: IIF()	(FP function)
Calls: EMPTY()	(FP function)
Calls: DELFILE()	(function in HMCOMP.SPR)
Contains: SETUPBOOT()	(Params: none)
Called by: _Q9BOS2HYD()	(function in HMCOMP.SPR)
Calls: ACOPY()	(FP function)
Calls: ADEL()	(FP function)
Calls: ALEN()	(FP function)
Calls: BSELECT()	(SELECT.PRG)
Calls: EMPTY()	(FP function)
Calls: ERRMSG()	(function in ERRMSG.PRG)
Calls: ASCAN()	(FP function)
Calls: ASUBSCRIPT()	(FP function)
Calls: BWFACT()	(function in BWFACT.PRG)
Calls: DP()	(function in DP.PRG)
Contains: BSTEP()	(Params: MAT, LCPHASE, WP, SN, SD, SQ)
Called by: COMPUTE	(procedure in HMCOMP.SPR)
Calls: N()	(FP function)
Calls: RESETWT()	(function in HMCOMP.SPR)
Calls: CALCULATE()	(function in HMCOMP.SPR)
Calls: STD()	(FP function)
Calls: AVG()	(FP function)
Calls: ALLTRIM()	(FP function)
Calls: STR()	(FP function)
Calls: DELFILE()	(function in HMCOMP.SPR)
Contains: BFACT()	(Params: M.FACTID)
Called by: COMPUTE	(procedure in HMCOMP.SPR)
Calls: PARAMETER()	(FP function)
Calls: SELECT()	(FP function)
Calls: N()	(FP function)
Calls: RESETWT()	(function in HMCOMP.SPR)
Calls: CALCULATE()	(function in HMCOMP.SPR)
Calls: STD()	(FP function)
Calls: AVG()	(FP function)
Calls: ALLTRIM()	(FP function)
Calls: STR()	(FP function)
Calls: DELFILE()	(function in HMCOMP.SPR)
Contains: BWSTEP()	(Params: M.FACTID, M.STEPID)
Called by: COMPUTE	(procedure in HMCOMP.SPR)
Calls: PARAMETER()	(FP function)
Calls: SELECT()	(FP function)
Calls: N()	(FP function)
Calls: RESETWT()	(function in HMCOMP.SPR)
Calls: CALCULATE()	(function in HMCOMP.SPR)
Calls: STD()	(FP function)
Calls: AVG()	(FP function)
Calls: ALLTRIM()	(FP function)

Calls: STR()	(FP function)	
Calls: DELFILE()	(function in HMCOMP.SPR)	
Contains: BPHASE()	(Params: M.PHASEID)	
Called by: COMPUTE	(procedure in HMCOMP.SPR)	
Calls: PARAMETER()	(FP function)	
Calls: SELECT()	(FP function)	
Calls: N()	(FP function)	
Calls: RESETWT()	(function in HMCOMP.SPR)	
Calls: CALCULATE()	(function in HMCOMP.SPR)	
Calls: STD()	(FP function)	
Calls: AVG()	(FP function)	
Calls: ALLTRIM()	(FP function)	
Calls: STR()	(FP function)	
Calls: DELFILE()	(function in HMCOMP.SPR)	
Contains: _Q9BOS2GG5()	Called by: HMCOMP.SPR	Contains:
_Q9BOS2GVA()	(Params: none)	
Called by: HMCOMP.SPR		
Contains: _Q9BOS2H5I()	Called by: HMCOMP.SPR	Contains:
_Q9BOS2HGT()	(Params: none)	
Called by: HMCOMP.SPR		
Contains: Q9BOS2HR2()	(Params: none)	
Called By: HMCOMP.SPR		
Contains: Q9BOS2HYD()	(Params: none)	
Called By: HMCOMP.SPR		
Calls: SETUPBOOT()	(function in HMCOMP.SPR)	
Calls: POPUPSHOW()	(function in HMSC.PRG)	
Calls: COMPUTE	(procedure in HMCOMP.SPR)	
Calls: POPUPHIDE()	(function in HMSC.PRG)	
Calls: HMBROWSE	(procedure in HMCOMP.SPR)	
15. DISPLAY.PRG -- Last updated: 09/09/92 at 9:02		
Contains: Q8X0JCQJY()	(Params: none)	
Called By: DISPLAY.PRG		
Calls: W_PRINT.SPR		
16. W_PRINT.SPR -- Last updated: 09/23/92 at 13:06		
Contains: PRINTFILE()	(Params: none)	
Called by: Q9BOS330A()	(W_PRINT.SPR)	
Calls: ALLTRIM()	(FP function)	
Calls: EMPTY()	(FP function)	
Contains: Q9BOS33H5()	(Params: none)	
Called By: W_PRINT.SPR		
Contains: Q9BOS330A()	(Params: none)	
Called By: W_PRINT.SPR		
Calls: PRINTFILE()	(W_PRINT.SPR)	
Contains: Q9BOS33W4()	Called by: W_PRINT.SPR	
Calls: OBJNUM()	(FP function)	
17. BACKUP.PRG -- Last updated: 07/15/92 at 7:36		
Contains: ERRHAND	(Params: none)	
Calls: ERRMSG()	(function in ERRMSG.PRG)	
Contains: STOP	(Params: none)	
Calls: EMPTY()	(FP function)	
Calls: ERROR()	(FP function)	
Calls: ERRMSG()	(function in ERRMSG.PRG)	



---

18. HMLU.PRG -- Last updated: 08/28/92 at 9:20

Contains: GET\_HMNAME() (Params: none)  
Called by: \_Q7D001E18() (function in HMLU.PRG)  
Calls: SELECT() (FP function)  
Calls: EMPTY() (FP function)  
Calls: CHOOSER() (CHOOSER.PRG)  
Calls: ERRMSG() (function in ERRMSG.PRG)  
Contains: Q7D001E18() (Params: none)  
Called By: HMLU.PRG  
Calls: EMPTY() (FP function)  
Calls: ALLTRIM() (FP function)  
Calls: IIF() (FP function)  
Calls: UPPER() (FP function)  
Calls: GET\_HMNAME() (function in HMTAB.SPR)  
Contains: Q7D001EUB() (Params: none)  
Called By: HMLU.PRG

19. MEMOWIN.PRG -- Last updated: 09/08/92 at 16:47

Contains: Q8L0LJ6F0() (Params: none)  
Called By: MEMOWIN.PRG  
Called by: MEMOEDIT.PRG  
Calls: W\_PRINT.SPR

20. HMSCEN.PRG -- Last updated: 08/27/92 at 8:42

Contains: GET\_HMSC() (Params: NAME)  
Called by: \_Q8KOINGJH() (function in HMSCEN.PRG)  
Called by: \_Q8KOINGTV() (function in HMSCEN.PRG)  
Calls: ALLTRIM() (FP function)  
Calls: EMPTY() (FP function)  
Calls: CHOOSER() (CHOOSER.PRG)  
Calls: ASCAN() (FP function)  
Calls: STR() (FP function)  
Contains: MEVENT (Params: MNEW)  
Called by: \_Q8KOINGJH() (function in HMSCEN.PRG)  
Called by: \_Q8KOINGTV() (function in HMSCEN.PRG)  
Called by: \_Q8KOINHVBV() (function in HMSCEN.PRG)  
Contains: Q8KOINGJH() (Params: none)  
Called By: HMSCEN.PRG  
Calls: EMPTY() (FP function)  
Calls: ALLTRIM() (FP function)  
Calls: IIF() (FP function)  
Calls: UPPER() (FP function)  
Calls: GET\_HMSC() (function in HMSCEN.PRG)  
Calls: MEVENT (procedure in HMSCEN.PRG)  
Calls: DP() (function in DP.PRG)  
Calls: VAL() (FP function)  
Contains: Q8KOINGTV() (Params: none)  
Called By: HMSCEN.PRG  
Calls: EMPTY() (FP function)  
Calls: OBJNUM() (FP function)  
Calls: ERRMSG() (function in ERRMSG.PRG)  
Calls: GET\_HMSC() (function in HMSCEN.PRG)  
Calls: MEVENT (procedure in HMSCEN.PRG)  
Calls: DP() (function in DP.PRG)  
Calls: VAL() (FP function)  
Contains: \_Q8KOINHVBV() (Params: none)

Called by: HMSCEN.PRG  
Calls: MEVENT

(procedure in HMSCEN.PRG)

21. HMSTEP.PRG -- Last updated: 09/09/92 at 9:21

Contains: ESCPRESSED	(Params: none)
Contains: INITVAR	(Params: none)
Called by: HMSTEP.PRG	
Called by: _Q8ROY3J3T()	(function in HMSTEP.PRG)
Called by: _Q8ROY3L13()	(function in HMSTEP.PRG)
Calls: GET_HMATN()	(function in HMSTEP.PRG)
Calls: SPACE()	(FP function)
Calls: GET_HMLCN()	(function in HMTAB.SPR)
Calls: GET_HMWPN()	(function in HMTAB.SPR)
Contains: ADDOPTION	(Params: none)
Called by: _Q8ROY3J3T()	(function in HMSTEP.PRG)
Called by: _Q8ROY3L13()	(function in HMSTEP.PRG)
Contains: CHANGE	(Params: none)
Called by: _Q9BORZPAH()	(function in HMAT.SPR)
Called by: _Q9BOS00OG()	(function in HMLC.SPR)
Called by: _Q9BOS0JCN()	(function in HMET.SPR)
Called by: _Q9BOS1HTG()	(function in HMCFEI.SPR)
Called by: _Q9BOS1ZXN()	(function in HMTAB.SPR)
Called by: _Q8ROY3L13()	(function in HMSTEP.PRG)
Calls: SET()	(FP function)
Calls: TRIM()	(FP function)
Calls: USED()	(FP function)
Calls: UPPER()	(FP function)
Calls: ALLTRIM()	(FP function)
Contains: GET_HMAT()	(Params: NAME)
Called by: _Q8ROY3IQX()	(function in HMSTEP.PRG)
Calls: SELECT()	(FP function)
Calls: EMPTY()	(FP function)
Calls: CHOOSER()	(CHOOSER.PRG)
Calls: ASCAN()	(FP function)
Calls: STR()	(FP function)
Calls: ERRMSG()	(function in ERRMSG.PRG)
Contains: GET_HMATN()	(Params: M.ID)
Called by: INITVAR	(procedure in HMSTEP.PRG)
Contains: GET_HMLC()	(Params: M.HMLC)
Called by: _Q9BOS1XI0()	(function in HMTAB.SPR)
Called by: _Q8ROY3JYQ()	(function in HMSTEP.PRG)
Calls: SELECT()	(FP function)
Calls: EMPTY()	(FP function)
Calls: CHOOSER()	(CHOOSER.PRG)
Calls: ERRMSG()	(function in ERRMSG.PRG)
Calls: ASCAN()	(FP function)
Calls: STR()	(FP function)
Contains: GET_HMLCN()	(Params: M.ID)
Called by: HMTAB.SPR	
Called by: _Q9BOS1VJD()	(function in HMTAB.SPR)
Called by: _Q9BOS1ZXN()	(function in HMTAB.SPR)
Called by: INITVAR	(procedure in HMSTEP.PRG)
Calls: SELECT()	(FP function)
Contains: GET_HMWP()	(Params: M.MATCH)
Called by: _Q9BOS1XZO()	(function in HMTAB.SPR)
Called by: _Q8ROY3KIH()	(function in HMSTEP.PRG)
Calls: SELECT()	(FP function)
Calls: EMPTY()	(FP function)

Calls: CHOOSE()	(CHOOSE.PRG)
Calls: ERRMSG()	(function in ERRMSG.PRG)
Calls: UPPER()	(FP function)
Calls: ALLTRIM()	(FP function)
Calls: ASCAN()	(FP function)
Calls: STR()	(FP function)
Contains: GET_HMWP()	(Params: M.ID)
Called by: HMTAB.SPR	
Called by: _Q9BOS1VJD()	(function in HMTAB.SPR)
Called by: _Q9BOS1ZXN()	(function in HMTAB.SPR)
Called by: INITVAR	(procedure in HMSTEP.PRG)
Calls: SELECT()	(FP function)
Contains: GET_HMUNIT	(Params: none)
Called by: _Q8ROY3IQX()	(function in HMSTEP.PRG)
Called by: _Q8ROY3JYQ()	(function in HMSTEP.PRG)
Called by: _Q8ROY3KIH()	(function in HMSTEP.PRG)
Calls: EMPTY()	(FP function)
Calls: ALN()	(FP function)
Calls: ERRMSG()	(function in ERRMSG.PRG)
Contains: SCSAVE	(Params: none)
Called by: _Q8ROY3J3T()	(function in HMSTEP.PRG)
Calls: SELECT()	(FP function)
Calls: SEEK()	(FP function)
Contains: DATACHECK()	(Params: none)
Called by: _Q8ROY3L13()	(function in HMSTEP.PRG)
Calls: EMPTY()	(FP function)
Calls: ERRMSG()	(function in ERRMSG.PRG)
Calls: OBJNUM()	(FP function)
Contains: _Q8ROY3IHW()	(Params: none)
Called By: HMSTEP.PRG	
Contains: _Q8ROY3IQX()	(Param.: none)
Called By: HMSTEP.PRG	
Calls: EMPTY()	(FP function)
Calls: ALLTRIM()	(FP function)
Calls: IIF()	(FP function)
Calls: UPPER()	(FP function)
Calls: GET_HMAT()	(function in HMSTEP.PRG)
Calls: DP()	(function in DP.PRG)
Calls: VAL()	(FP function)
Calls: SPACE()	(FP function)
Calls: GET_HMUNIT	(procedure in HMSTEP.PRG)
Contains: _Q8ROY3J3T()	(Params: none)
Called By: HMSTEP.PRG	
Calls: IIF()	(FP function)
Calls: RECCOUNT()	(FP function)
Calls: RECNO()	(FP function)
Calls: YESNO()	(function in YESNO.PRG)
Calls: OBJNUM()	(FP function)
Calls: INITVAR	(procedure in HMSTEP.PRG)
Calls: ADDOPTION	(procedure in HMSTEP.PRG)
Calls: EOF()	(FP function)
Calls: CHR()	(FP function)
Calls: BOF()	(FP function)
Calls: SCSAVE	(procedure in HMSTEP.PRG)
Contains: _Q8ROY3JSG()	(Params: none)
Called By: HMSTEP.PRG	
Contains: _Q8ROY3JYQ()	(Params: none)
Called By: HMSTEP.PRG	
Calls: EMPTY()	(FP function)

Calls: ALLTRIM()	(FP function)
Calls: IIF()	(FP function)
Calls: UPPER()	(FP function)
Calls: GET_HMLC()	(function in HMTAB.SPR)
Calls: DP()	(function in DP.PRG)
Calls: VAL()	(FP function)
Calls: SPACE()	(FP function)
Calls: GET_HMUNIT	(procedure in HMSTEP.PRG)
Contains: Q8ROY3KC7()	(Params: none)
Called By: HMSTEP.PRG	
Contains: Q8ROY3KIH()	(Params: none)
Called By: HMSTEP.PRG	
Calls: EMPTY()	(FP function)
Calls: ALLTRIM()	(FP function)
Calls: IIF()	(FP function)
Calls: UPPER()	(FP function)
Calls: GET_HMWP()	(function in HMTAB.SPR)
Calls: DP()	(function in DP.PRG)
Calls: VAL()	(FP function)
Calls: SPACE()	(FP function)
Calls: GET_HMUNIT	(procedure in HMSTEP.PRG)
Contains: Q8ROY3L13()	(Params: none)
Called By: HMSTEP.PRG	
Calls: DATACHECK()	(function in HMSTEP.PRG)
Calls: CHANGE	(procedure in HMTAB.SPR)
Calls: RECNO()	(FP function)
Calls: RECCOUNT()	(FP function)
Calls: EMPTY()	(FP function)
Calls: YESNO()	(function in YESNO.PRG)
Calls: INITVAR	(procedure in HMSTEP.PRG)
Calls: ADDOPTION	(procedure in HMSTEP.PRG)
Contains: Q8ROY3LQZ()	(Params: none)
Called By: HMSTEP.PRG	

22. MEMOEDIT.PRG -- Last updated: 09/09/92 at 9:19  
 Contains: Q8L0LJ6F0() (Params: none)  
 Called By: MEMOWIN.PRG  
 Called by: MEMOEDIT.PRG  
 Calls: W\_PRINT.SPR
23. CHOOSER.PRG -- Last updated: 07/16/92 at 13:06  
 Contains: Q120IDFTF() (Params: none)  
 Called By: CHOOSER.PRG  
 Contains: Q120IDG17() (Params: none)  
 Called By: CHOOSER.PRG  
 Contains: Q120IDG86() (Params: none)  
 Called By: CHOOSER.PRG  
 Calls: OBJNUM() (FP function)
24. BSELECT.PRG -- Last updated: 09/02/92 at 10:49  
 Contains: Q8QON6SBM() (Params: none)  
 Called By: BSELECT.PRG
25. BWFACT.PRG -- Last updated: 09/02/92 at 11:04  
 Contains: Q8QONQKA6() (Params: none)  
 Called By: BWFACT.PRG

---

## **Section VII. Program Source Code**

```

*****
1  * Procedure file: C:\MHLCOH\WORK\BACKUP.PRG
2  *
3  *
4  *
5  * System: Hazardous Material Life-Cycle Cost
6  * Author: Ly, Hoa
7  * Copyright (c) SEPT. 1992, Naval Health Research Center
8  * Last modified: 07/15/92 7:36
9  *
10 * Procs & Frcts: BACKUP()
11 *                : ERRHAND
12 *                : STOP
13 *
14 * Calls: ERRMSG() (function in ERRMSG.PRG)
15 *
16 * Documented 10/28/92 at 13:48 FoxDoc version 2.10
17 *
18 * set talk off
19 * errmsg("insert diskette in drive A and hit RETURN")
20 * errmsg("please wait.",1)
21 *
22 * errmsg("DONE.",1)
23 *
24 * return
25 *
26 *
27 *
28 * Procedure: ERRHAND
29 *
30 * Calls: ERRMSG() (function in ERRMSG.PRG)
31 *
32 *
33 * procedure errhand
34 *   errmsg("BACKUP CRASHED CHECK PARAMETERS AND TRY AGAIN.")
35 *   return
36 *   quit
37 *
38 *
39 *
40 *
41 *
42 * Procedure: STOP
43 *
44 * Calls: EMPTY() (function in ?)
45 *        : ERROR() (function in ?)
46 *        : ERRMSG() (function in ERRMSG.PRG)
47 *
48 *
49 * procedure stop
50 *   if empty(error())
51 *     errmsg("please check drive A")
52 *     retry
53 *   endif
54 *   return
55 * * EOF: BACKUP.ACT
56

```

```

1 *****
2
3 Procedure file: C:\WWW.CCM\WORK\YESNO.PRG
4
5 System: Hazardous Material Life-Cycle Cost
6 Author: LY, Hoe
7 Copyright (c) SEP 1992, Naval Health Research Center
8 Last modified: 07/07/92 8:16
9
10 Proc & Fncts: YESNO(X)
11
12 Calls: MESSAGE() (function in ?)
13 : ERRMSG() (function in ?)
14 : PARAMETERS() (function in ?)
15 : IIF() (function in ?)
16 : TYPE() (function in ?)
17 : LEN() (function in ?)
18 : SUBSTR() (function in ?)
19 : UPPER() (function in ?)
20 : SET() (function in ?)
21 : WEXIST() (function in ?)
22 : INT() (function in ?)
23 : SROW() (function in ?)
24 : SCOL() (function in ?)
25 : WVISIBLE() (function in ?)
26 : PAD() (function in ?)
27 : LASTKEY() (function in ?)
28
29 Documented 10/28/92 at 13:48 FoxDoc version 2.10
30 *****
31 * * * YESNO.PRG 22:21:11 * * *
32 * * *
33 * * *
34 * * *
35 * * *
36 * * *
37 * * * Description:
38 * * * This program was automatically generated by GENSCRN.
39 * * *
40 * * *
41 * * *
42 * * *
43 * * *
44 * * *
45 * * * YESNO Setup Code - SECTION 1
46 * * *
47 * * *
48 * * *
49
50 #region 1
51 parameters message, ok, cancel
52 private all
53
54 on error do errmsg with message()
55
56 * If no message is sent then create message
57
58
59
60
61
62
63
64
65
66

```

```

67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132

```

```

m.ok = "OK"
m.cancel = "Cancel"
endif

* Truncate any message longer than 50 characters
message=iif(type(m.message)="C",m,m.message)
if len(m.message) > 50
* The message is centered in an ASAY
m.message = substr(m.message, 1, 50)
endif

m.message=upper(m.message)
m.ok=upper(m.ok)
m.cancel=upper(m.cancel)

push key clear

#region 0
regional m.curraera, m.talkstat, m.compatat
set talk off
m.talkstat=set("TALK")
set compatible foxplus

m.compatat = set("COMPATIBLE")
set compatible foxplus

*****
* * * Window definitions
* * *
*****

if not wexist("yesno")
define window yesno ;
from int((srow()-7)/2),int((scol()-48)/2) ;
to int((srow()-7)/2)+6,int((scol()-48)/2)+47 ;
float ;
noclose ;
shadow ;
double ;
color scheme 5
endif

*****
* * * YESNO Screen Layout
* * *
*****

#region 1
if wvisible("yesno")
activate window yesno same
else
activate window yesno noshw
endif
a 1,2 say padc(m.message, 40) ;
size 1,41
a 3,8 get m.answer ;
picture "g=HT \\\<OK;\<Cancel" ;
size 1,12.5 ;
default 1

if not wvisible("yesno")

```

```

133 activate window yesno
134 _endif
135
136 read cycle modal
137
138 release window yesno
139
140 #region 0
141
142 _if m.talkstat = "ON"
143   set talk on
144 _else
145   set talk off
146 _endif
147
148 _if m.compatstat = "ON"
149   set compatible on
150 _endif
151
152 * *****
153 * *
154 * * YESNO Cleanup Code
155 * *
156 * *
157 * * *****
158
159 #region 1
160 pop key
161
162 * Convert the numeric value of m.answer from 1|2 to .T|.F.
163 * If the user selected OK and didn't exit with Escape
164 _if m.answer = 1 and lastkey() <> 27
165   return .t.
166 _else
167   * Cancel or Escape returns false
168   return .f.
169 _endif
170 *: EOF: YESNO.ACT
171

```







1	*	08/28/92	memowin.PRG	10:02:45	
2	*				
3	*				
4	*				
5	*				
6	*				
7	*				
8	*				
9	*				
10	*	Author's Name			
11	*	Copyright (c) 1992 Company Name			
12	*	Address			
13	*	City, Zip			
14	*	Descriptions:			
15	*	This program was automatically generated by GENSCRN.			
16	*				
17	*				
18	*				
19	*				
20	*				
21	*				
22	*				
23	*				
24	*				
25	*				
26	*				
27	*				
28	*				
29	*				
30	*				
31	*				
32	*				
33	*				
34	*				
35	*				
36	*				
37	*				
38	*				
39	*				
40	*				
41	*				
42	*				
43	*				
44	*				
45	*				
46	*				
47	*				
48	*				
49	*				
50	*				
51	*				
52	*				
53	*				
54	*				
55	*				
56	*				
57	*				
58	*				
59	*				
60	*				
61	*				
62	*				
63	*				
64	*				
65	*				
66	*				
67	*				
68	*				
69	*				
70	*				
71	*				
72	*				
73	*				
74	*				
75	*				
76	*				
77	*				
78	*				
79	*				
80	*				
81	*				
82	*				
83	*				
84	*				
85	*				
86	*				
87	*				
88	*				
89	*				
90	*				
91	*				
92	*				
93	*				
94	*				
95	*				
96	*				
97	*				
98	*				
99	*				
100	*				
101	*				
102	*				
103	*				
104	*				
105	*				
106	*				
107	*				
108	*				
109	*				
110	*				
111	*				
112	*				
113	*				
114	*				
115	*				
116	*				
117	*				
118	*				
119	*				
120	*				
121	*				
122	*				
123	*				
124	*				
125	*				
126	*				
127	*				
128	*				
129	*				
130	*				
131	*				
132	*				

```
133 *
134 *
135 *
136 *
137 *
138 *
139 *
140 *
141 *
142 *
143 *
144 *
145 *
146 *
147 *
148 *
149 *
150 *
151 *
153 *
```

Function Origin: 4

From Screen: memwin, Record Number: 4

Variable: m.save

Called By: VALID Clause

Object Type: Push Button

Snippet Number: 1

```
function _q8l0lj6f0 22 m.save VALID
if m.save = 1
do w_print.spr with m.text
endif
*: EOF: MEMOIN.ACT
```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

```

06/28/92	MEMOEDIT.PRG	10:02:45
Hoe L. Ly Copyright (c) 1992 Naval Health Research Center P..Box 85122 San Diego, CA 92186-5122 Description: This program was automatically generated by GENSCRN.		
MEMOEDIT Setup Code - SECTION 1		
<pre> region 1 parameter text, title, editallow private all do case   m.text = ""   m.title = ""   m.editallow = .t. case parameter() = 1   m.title = ""   m.editallow = .t. case parameter() = 2   m.editallow = .t. endcase </pre>		
<pre> region 0 regional m.curreas, m.talkstat, m.compatat if set("TALK") = "ON"   set talk off   m.talkstat = "ON" else   m.talkstat = "OFF" endif m.compatat = set("COMPATIBLE") set compatible foxplus m.curreas = select() </pre>		
Window definitions		
<pre> if not exist("memoedit")   define window memoedit ;     from int((row()-20)/2),int((col()-77)/2) ;     to int((row()-20)/2)+16,int((col()-77)/2)+76 ; </pre>		

```

133 release window memoedit
134 select (m.currarea)
135
136
137
138 #region 0
139 if m.talkstat = "ON"
140     set talk on
141 endif
142 if m.comstat = "ON"
143     set compatible on
144 endif
145
146
147
148
149
150
151
152
153
154 #region 1
155 return m.text
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172 function _q8l0lJ6f0 22 m.save VALID
173 #region 1
174 if m.save = 2
175     m.text = m.oldtext
176 endif
177
178 *: EOF: MEMOEDIT.ACT

```

## MEMOEDIT Cleanup Code

_q8l0lJ6f0	m.save VALID	
Function Origin:	MEMOEDIT,	Record Number: 4
From Screen:	m.save	
Variable:	VALID Clause	
Called By:	Push Button	
Object Type:	1	
Snippet Number:		

10/09/92	HMWP.SPR	15:38:39
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCREEN.		

```

1 *
2 *
3 *
4 *
5 *
6 *
7 *
8 *
9 *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *

```

```

#region 0
regional m.curreas, m.talkstat, m.compatstat

```

```

  if set("TALK") = "ON"
    set talk off
    m.talkstat = "ON"
  else
    m.talkstat = "OFF"
  endif
  m.compatstat = set("COMPATIBLE")
  set compatible foxplus

```

#### Window definitions

```

  if not wxstat("hmap")
    define window hmap
    from int((row()-15)/2), int((col()-64)/2)
    to int((row()-15)/2)+14, int((col()-64)/2)+63
    nofloat
    noclose
    shadow
    double
    color scheme 1
  endif

```

#### HMWP Setup Code - SECTION 2

```

#region 1
push key
  ON KEY LABEL ESC DO EscPressed
  m.oldscape = set( "ESCAPE" )
  set escape off
  m.adding = .f.
  m.change = .f.
endif
select 0

```

```

67 use hmap
68 set order to tag hmap of hmap.cdx
69
70 else
71 select hmap
72 set order to tag hmap
73 endif
74
75 *****
76 * Check see if the last record is defined
77 if type( "m.LastRec" ) = "U"
78
79 * Start with the first record
80 go top
81 m.lastrec = recno()
82
83 else
84 * Start on the last record used
85 go m.lastrec
86 endif
87 *****
88 scatter memvar
89
90 *
91 *
92 *
93 *
94 *
95 *
96
97 #region 1
98 if wvisible("hmap")
99 activate window hmap same
100
101 else
102 activate window hmap noshow
103 endif
104
105 @ 2,50 get m.action
106 picture "g*VN \<add;\<edit;\<next;\<previous;\<7E\<xit"
107 size 1,10,1
108 default 1
109 valid _qpr0xj4qy()
110 @ 2,11 get m.hmapid
111 size 1,10
112 default " "
113 disable
114 @ 4,11 get m.hmap
115 size 6,34
116 default " "
117 picture "aj"
118 valid _qpr0xj5b7()
119 disable
120 @ 11,14 get m.save
121 picture "g*HN \<save;\<cancel"
122 size 1,8,1
123 default 1
124 valid _qpr0xj5l1()
125 disable
126 @ 1,0 to 12,46
127 @ 2,3 say "ID NUM:"
128 @ 4,2 say "PROCESS:"
129 @ 0,16 say "HM PROCESSES"
130
131 if not wvisible("hmap")
132 activate window hmap
133 endif

```

#### HMWP Screen Layout

```

133 read cycle
134
135 release window hmap
136
137 #region 0
138 if m.talkstat = "ON"
139 set talk on
140 endif
141 if m.compatat = "ON"
142 set compatible on
143 endif
144
145 *
146 *
147 *
148 *
149 *
150 *
151
152 #region 1
153 if used('HMWP')
154 use
155 endif
156 pop key all
157 set escape &oldescape
158 ***** End of Main Body - Entry Cleanup
159
160 *****
161 procedure change
162 *****
163 m.oldexact = set( "EXACT" )
164 set exact on
165 m.change &trim(hmap.hmap) <> trim( m.hmap )
166 set exact &oldexact
167 return m.change
168
169 *
170 *
171 *
172 *
173 *
174 *
175 *
176 *
177 *
178 *
179 *
180 *
181 *
182 *
183
184 function q9r0xj4qy 22 m.Action VALID
185 #region 1_q9r0xj4qy
186 if m.action = 1
187 scatter memvar blank
188 m.hmapid=recno()+1
189 show gets
190 show get m.hmap enabled
191 show get action disabled
192 show get save enabled
193 m.adding = .t.
194
195 -else
196 -do case
197
198

```

## HMWP Cleanup Code

```

_Q9R0XJ4QY      m.Action VALID
Function Origin:
From Screen:    HMWP,      Record Number: 2
Variable:       m.Action
Called By:      VALID Clause
Object Type:    Push Button
Snippet Number: 1

```

```

199 -case m.action = 2
200 show gets
201 show get m.hmap enabled
202 show get action disabled
203 show get save enabled
204
205 -case m.action = 3
206 skip
207 if eof()
208 ?? chr( 7 )
209 wait "Last record" window nowait
210 skip -1
211 -else
212 scatter memvar
213 show gets
214 -endif
215
216 -case m.action = 4
217 skip -1
218 if bof()
219 ?? chr( 7 )
220 wait "First record" window nowait
221 skip
222 -else
223 scatter memvar
224 show gets
225 -endif
226
227 -case m.action = 5
228 clear read
229 -endcase
230
231 -endif
232
233 *
234 *
235 *
236 *
237 *
238 *
239 *
240 *
241 *
242 *
243 *
244 *
245 *
246
247 function q9r0xj5b7 22 m.hmap VALID
248 #region 1_q9r0xj5b7
249 if m.adding
250 m.oldrec = recno()
251 seek m.hmap
252 if found()
253 -errmsg("Record already exists",1)
254 scatter field hmap memvar blank
255 curobj = objnum(m.hmap)
256 -endif
257 go m.oldrec
258 -endif
259 show gets
260
261 *
262 *
263 *
264

```

```

_Q9R0XJ5B7      m.hmap VALID
Function Origin:
From Screen:    HMWP,      Record Number: 4
Variable:       m.hmap
Called By:      VALID Clause
Object Type:    Field
Snippet Number: 2

```

```

_Q9R0XJ5L1      m.Save VALID

```



```

265 *
266 *
267 *
268 *
269 *
270 *
271 *
272 *
273 *
274 *
275 *
276 *
277 *
278 *
279 *
280 *
281 *
282 *
283 *
284 *
285 *
286 *
287 *
288 *
289 *
290 *
291 *
292 *
293 *
294 *
295 *
296 *
297 *
298 *
299 *
300 *
301 *
302 *
303 *
304 *
305 *
306 *
307 *
308 *
309 *
310 *
311 *
312 *

```

Function Origin:

From Screen: IHMAP, Record Number: 5

Variable: m.Save

Called By: VALID Clause

Object Type: Push Button

Snippet Number: 3

```

function qprOxjS11 22 m.Save VALID
#region 1
m.notesave = .f.
if m.save = 1 22 Selected Save Button
    if empty(m.heap)
        errmsg("Data empty, could not save!!",2)
        curobj = objnum(m.heap)
        m.notesave = .t.
    else
        if m.adding 22 Adding a new record
            append blank
            gather memvar
        else
            if m.change 22 Changing an old record
                gather memvar
            endif
        endif
    endif
    scatter memvar
endif

show gets
if m.notesave
    show get m.heap enabled
    show get action disabled
    show get save enabled
else
    show get m.heap disabled
    show get action enabled
    show get save disabled
    m.adding = .f.
    m.change = .f.
endif

```

\*: EOF: IHMAP.AC1

10/15/92	HMTAB.SPR	10:14:49
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCREEN.		

```

1 *
2 *
3 *
4 *
5 *
6 *
7 *
8 *
9 *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *

```

```

18 #region 0
19 regional m.curarea, m.talkstat, m.compat

```

```

20 -if set("TALK") = "ON"
21   set talk off
22   m.talkstat = "ON"
23 -else
24   m.talkstat = "Off"
25 -endif
26
27 m.compat = set("COMPATIBLE")
28 set compatible foxplus
29
30
31

```

### Window definitions

```

32 *
33 *
34 *
35 *
36 *
37
38 -if not exist("hmtab")
39   define window hmtab ;
40   from int((row()-22)/2), int((col()-79)/2) ;
41   to int((row()-22)/2)+21, int((col()-79)/2)+78 ;
42   float ;
43   noclose ;
44   shadow ;
45   double ;
46   color scheme 1
47 -endif
48
49
50
51
52
53
54
55

```

### HMTAB Setup Code - SECTION 2

```

56 #region 1
57 push key
58 "ON KEY LABEL ESC DO EscPressed
59 m.oidescape = set("ESCAPE")
60 set escape off
61 m.adding = .f.
62 m.change = .f.
63 m.action=5
64
65
66

```

```

67 *****
68 * SELECT HAZARDOUS MATERIAL
69 *****
70 m.hname=hlu()
71 -if empty(m.hname)
72   -errmsg("No material selected",1)
73   -return
74 -endif
75
76 close all
77 *****
78 * GET UNIT LIST
79 *****
80 -if used("HUNIT")
81   select 0
82   use hunit order tag hunit
83 -else
84   select hunit
85 -endif
86 copy to array unitlist field hunit
87 use
88 m.unit = "GALLON"
89 *****
90 * GET THE MATERIAL RECORDS
91 *****
92 m.hmatid=get_hmid(m.hname)
93 select 0
94 use hmtab alias hmtab
95 set filter to hmatid = m.hmatid
96
97
98 *What if the file is empty?
99
100 go top
101 m.oidhmatid=m.hmatid
102 scatter memvar
103 m.oidunit = m.hunit
104
105 -if m.tabid<1
106   m.action=1
107   m.adding=.t.
108   m.tabid=reccount()+1
109   m.hmatid=m.oidhmatid
110   m.hmlc=""
111   m.hmlcid=0
112   m.oidhmlc=""
113   m.hmet=""
114   m.oidhmet=""
115   m.hmetid=0
116   m.hmetprob=0.000
117   m.hmpa=""
118   m.oidhmpa=""
119   m.hmwpid=0
120   m.hmcf=""
121   m.hmcfid=0
122   m.oidhmcf=""
123   m.hmcfid=0
124   m.hmcfes=""
125   m.oidhmcfes=""
126   m.hmcfid=0
127   m.hmcfes=""
128   m.hmcfecost=0.00
129   m.hunit=if(empty(m.oidunit),"GALLON",m.oidunit)
130   m.proba=0.00
131   m.wtagerage=0.00
132

```

```

133 else
134
135     if m.halcid>0
136         m.halc=get_halc(m.halcid)
137         m.olchalc=m.halc
138     else
139         m.halc=""
140         m.olchalc=""
141     endif
142
143     if m.hmapid>0
144         m.hmap=get_hmap(m.hmapid)
145         m.olchmap=m.hmap
146     else
147         m.hmap=""
148         m.olchmap=""
149     endif
150
151     if m.hmetid>0
152         m.hmet=get_hmet(m.hmetid)
153         m.olchmet=m.hmet
154     else
155         m.hmet=""
156         m.olchmet=""
157     endif
158
159     if m.hmcfid>0
160         m.hmc=get_cfar(m.hmcfid)
161         m.olchmc=m.hmc
162     else
163         m.hmc=""
164         m.olchmc=""
165     endif
166
167     if m.hmcfid>0
168         m.hmcf=get_cfar1(m.hmcfid)
169         m.olchmcf=m.hmcf
170     else
171         m.hmcf=""
172         m.olchmcf=""
173     endif
174
175     if m.hmcfelid>0
176         m.hmcfel=get_el(m.hmcfelid)
177         m.olchmcfel=m.hmcfel
178     else
179         m.hmcfel=""
180         m.olchmcfel=""
181     endif
182
183     show gets
184
185     *
186     *
187     *
188     *
189     *
190     *
191     *
192     *
193     *
194     *
195     #region 1
196     if visible("hmtab")
197         activate window hmtab same
198
199     else
200         activate window hmtab noshow
201     endif
202
203     @ 1,67 get m.action ;
204     picture "g"VW \<Add;\<Edit;\<Delete;\<Previous;\<Top;\<Bottom;\<VE
205     =>\<xit"" ;
206
207     size 1,10,1 ;
208     default 1 ;
209     valid qp00lyrbv() ;
210     @ 0,5 get m.tabid ;
211     size 1,4 ;
212     default 0 ;
213     disable
214     @ 1,14 get m.hmetid ;
215     size 1,4 ;
216     default 0 ;
217     picture "g2" ;
218     disable
219     @ 1,19 get m.hname ;
220     size 3,21 ;
221     default " " ;
222     picture "g1" ;
223     disable
224     @ 1,47 get m.hmunit ;
225     picture "g" ;
226     from unitlist ;
227     size 3,18 ;
228     default 1 ;
229     disable ;
230     color scheme 1, 2
231     @ 4,14 get m.halcid ;
232     size 1,4 ;
233     default 0 ;
234     picture "g2" ;
235     disable
236     @ 4,19 get m.halc ;
237     size 1,46 ;
238     default " " ;
239     picture "g1" ;
240     when qp00lyrbv() ;
241     valid qp00lyrbz() ;
242     disable
243     @ 5,13 get m.hmpid ;
244     size 1,5 ;
245     default 0 ;
246     picture "g2" ;
247     disable
248     @ 5,19 get m.hmap ;
249     size 2,46 ;
250     default " " ;
251     picture "g1" ;
252     when qp00lyrbv() ;
253     valid qp00lyrbz() ;
254     disable
255     @ 7,18 get m.hmetid ;
256     size 1,4 ;
257     default 0 ;
258     picture "g2" ;
259     disable
260     @ 7,24 get m.hmet ;
261     size 1,41 ;
262     default " " ;
263     picture "g1" ;
264     when qp00lyrbv() ;
265     valid qp00lyrbz() ;
266     disable

```

HMTAB Screen Layout



```

396 else
397   select 0
398   use hmtab alias hmtab
399   set filter to hmatid = m.hmatid
400   endif
401
402   m.oldexact = set( "EXACT" )
403   set exact on
404   m.change = (hmtab.hmatid <> m.hmatid;
405   or hmtab.hmatid <> m.hmatid;
406   or hmtab.hmatid <> m.hmatid;
407   or hmtab.hmatid <> m.hmatid;
408   or hmtab.hmatid <> m.hmatid;
409   or hmtab.hmatid <> m.hmatid;
410   or hmtab.hmatid <> m.hmatid;
411   or hmtab.hmatid <> m.hmatid;
412   or hmtab.hmatid <> m.hmatid;
413   or hmtab.hmatid <> m.hmatid;
414   or hmtab.hmatid <> m.hmatid;
415   or hmtab.hmatid <> m.hmatid;
416   or hmtab.hmatid <> m.hmatid;
417   or hmtab.hmatid <> m.hmatid;
418   or hmtab.hmatid <> m.hmatid;
419   or hmtab.hmatid <> m.hmatid;
420   or hmtab.hmatid <> m.hmatid;
421   or hmtab.hmatid <> m.hmatid;
422   or hmtab.hmatid <> m.hmatid;
423   or hmtab.hmatid <> m.hmatid;
424   or hmtab.hmatid <> m.hmatid;
425   or hmtab.hmatid <> m.hmatid;
426   or hmtab.hmatid <> m.hmatid;
427   or hmtab.hmatid <> m.hmatid;
428   or hmtab.hmatid <> m.hmatid;
429   or hmtab.hmatid <> m.hmatid;
430   or hmtab.hmatid <> m.hmatid;
431   or hmtab.hmatid <> m.hmatid;
432   or hmtab.hmatid <> m.hmatid;
433   or hmtab.hmatid <> m.hmatid;
434   or hmtab.hmatid <> m.hmatid;
435   or hmtab.hmatid <> m.hmatid;
436   or hmtab.hmatid <> m.hmatid;
437   or hmtab.hmatid <> m.hmatid;
438   or hmtab.hmatid <> m.hmatid;
439   or hmtab.hmatid <> m.hmatid;
440   or hmtab.hmatid <> m.hmatid;
441   or hmtab.hmatid <> m.hmatid;
442   or hmtab.hmatid <> m.hmatid;
443   or hmtab.hmatid <> m.hmatid;
444   or hmtab.hmatid <> m.hmatid;
445   or hmtab.hmatid <> m.hmatid;
446   or hmtab.hmatid <> m.hmatid;
447   or hmtab.hmatid <> m.hmatid;
448   or hmtab.hmatid <> m.hmatid;
449   or hmtab.hmatid <> m.hmatid;
450   or hmtab.hmatid <> m.hmatid;
451   or hmtab.hmatid <> m.hmatid;
452   or hmtab.hmatid <> m.hmatid;
453   or hmtab.hmatid <> m.hmatid;
454   or hmtab.hmatid <> m.hmatid;
455   or hmtab.hmatid <> m.hmatid;
456   or hmtab.hmatid <> m.hmatid;
457   or hmtab.hmatid <> m.hmatid;
458   or hmtab.hmatid <> m.hmatid;
459   or hmtab.hmatid <> m.hmatid;
460   or hmtab.hmatid <> m.hmatid;
461   or hmtab.hmatid <> m.hmatid;

```

```

462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527

```

```

*****
*GET LIFE CYCLE PHASE
function get hmlc
*****
parameter m.hmlc

release hmlc
dimension hmlc[1]
hmlc[1]=""
m.oldfile=select()

select hmlc.hmlc;
from hmlc;
where hmlc.hmlc in (alltrim(m.hmlc));
order by hmlc;
into array hmlc

m.ans=""
if not empty(hmlc[1])
  m.ans=chooser(hmlc,"Select a Life Cycle Phase")
else
  zerrmsg(m.hmlc + " was not found",1)
  m.ans = ""
endif

select (m.oldfile)
return m.ans

*****
function get hmlcn
parameter m.hmlcn
m.oldfile=select()

select distinct hmlc.hmlc;
from hmlc;
where hmlc.hmlc = (m.hmlcn);
into array x

m.hmlc=x[1]
select (m.oldfile)
return m.hmlc

*****
*Working Processes
function get hmlp
*****
parameter m.hmlp
release hmlp
dimension hmlp[1]
hmlp[1]=""
m.oldfile=select()

select hmlp.hmlp;
from hmlp;
where hmlp.hmlp in (alltrim(m.hmlp));
into array hmlp

m.ans=""
if not empty(hmlp[1])
  m.ans=chooser(hmlp,"Select a Process")
else
  zerrmsg(m.hmlp + " was not found",1)
  m.ans = ""
endif

```

```

528 select (m.oldfile)
529
530 return m.ans
531
532 *****
533 function get_hmapn
534 parameter m.hmapid
535 m.oldfile=select()
536
537 select distinct hmap.hmap;
538 from hmap;
539 where hmap.hmapid = m.hmapid;
540 into array x
541
542 m.hmap=x[1]
543 select(m.oldfile)
544 return m.hmap
545
546 *****
547 function get_hmet
548 *****
549 parameter m.hmet
550
551 release hmet
552 dimension hmet[1]
553 hmet[1]=""
554 m.oldfile=select()
555
556 select hmet.hmet;
557 from hmet;
558 where hmet.hmet in (alltrim(m.hmet));
559 into array hmet
560
561 m.ans=""
562 if not empty(hmet[1])
563 m.ans=chooser(hmet,"Select an Exposure Type")
564 else
565 m.ans="m.hmet + " was not found",1)
566 -endif
567
568 select (m.oldfile)
569 return m.ans
570
571 *****
572 function get_hmetid
573 *****
574 parameter m.hmet
575 m.oldfile=select()
576
577 select distinct hmet.hmetid;
578 from hmet;
579 where hmet.hmet in (alltrim(m.hmet));
580 into array x
581
582 m.hmetid=x[1]
583 select (m.oldfile)
584 return m.hmetid
585
586 *****
587 function get_hmetn
588 *****
589 parameter m.hmetid
590 m.oldfile=select()
591
592 select distinct hmet.hmet;
593 from hmet;

```

```

594 where hmet.hmetid = (m.hmetid);
595 into array x
596
597 m.hmet=x[1]
598 select (m.oldfile)
599 return m.hmet
600
601 ***** COST FACTORS *****
602
603 *****
604 function get_cfar
605 *get the cost factors
606 *****
607 parameter m.hmcfid
608 m.oldfile=select()
609
610 declare cfarr[1]
611 cfarr[1]=""
612
613 select hmcfcfarr[1]
614 from hmcfcf;
615 where hmcfcf.hmcfcid = (m.hmcfcid);
616 into array cfarr
617
618 m.hmcfcfarr[1]
619 select(m.oldfile)
620 return m.hmcfcf
621
622 *****
623 function get_cfar1
624 *get cost factor elements
625 *****
626 parameter m.hmcfcid
627 m.oldfile=select()
628 declare cfarr[1]
629 cfarr[1]=""
630
631 select hmcfcf.hmcfc;
632 from hmcfcf;
633 where hmcfcf.hmcfcid = (m.hmcfcid);
634 into array cfarr1
635
636 m.hmcfcf=cfarr1[1]
637 select (m.oldfile)
638 return m.hmcfcf
639
640 *****
641 function get_cf
642 *****
643 parameter m.hmcfcf
644 release memo like cfarr
645 dimension cfarr[1]
646 cfarr[1]=""
647
648 m.oldfile=select()
649
650 select hmcfcf.hmcfc;
651 from hmcfcf;
652 where hmcfcf.hmcfc in (alltrim(m.hmcfc));
653 order by hmcfcf.hmcfcid;
654 into array cfarr
655
656 if not empty(cfarr[1])
657
658
659

```

```

660 m.x=chooser(@cfearr1,"Select a Cost Factor")
661 -else
662   =errmsg(m.x + " was not found",1)
663   m.hmcfid=0
664   m.x=""
665 -endif
666
667 select (m.oldfile)
668 return m.x
669
670 *****
671 function get hmcfid
672 *****
673 parameter m.hmcfe
674 m.oldfile = select()
675 select distinct hmcfe.hmcfid;
676 from hmcfe;
677 where hmcfe.hmcfe in (alltrim(m.hmcfe));
678 into array x
679 m.hmcfid=x[1]
680 select (m.oldfile)
681 return m.hmcfid
682
683 *****
684 function get cf1
685 *****
686 parameter m.hmcfe
687
688 release memo like cfearr1
689 dimension cfearr1[1]
690 cfearr1[1]=""
691 m.oldfile=select()
692
693 select hmcfe.hmcfe;
694 from hmcfe;
695 where hmcfe.hmcfe in (alltrim(m.hmcfe));
696 order by hmcfe.hmcfe;
697 into array cfearr1
698
699 if not empty(cfearr1[1])
700 m.x=chooser(@cfearr1,"Select a Cost Factor Element")
701 -else
702   =errmsg(m.x + " was not found",1)
703   m.hmcfid=0
704   m.x=""
705 -endif
706 select (m.oldfile)
707 return m.x
708
709 *****
710 function get eid
711 *****
712 parameter m.hmcfe
713 m.oldfile = select()
714 select distinct hmcfe.hmcfe;
715 from hmcfe;
716 where hmcfe.hmcfe in (alltrim(m.hmcfe));
717 into array x
718 m.hmcfeid=x[1]
719
720
721
722
723
724
725

```

```

726 select (m.oldfile)
727 return m.hmcfeid
728
729 *****
730 function get ej
731 *****
732 parameter m.hmcfeid
733 m.oldfile=select()
734
735 select hmcfe.hmcfe;
736 from hmcfe;
737 where hmcfe.hmcfeid = (m.hmcfeid);
738 into array x
739 m.hmcfe=x[1]
740 select (m.oldfile)
741 return m.hmcfe
742
743 *****
744 function get elid
745 *****
746 parameter m.hmcfe
747 m.oldfile=select()
748
749 select hmcfe.hmcfeid;
750 from hmcfe;
751 where hmcfe.hmcfe in (alltrim(m.hmcfe));
752 into array x
753 m.hmcfeid=x[1]
754 select (m.oldfile)
755 return m.hmcfeid
756
757 *****
758 function rel
759 *****
760 parameter m.id
761 external array cost
762 m.oldfile=select()
763
764 if parameter()=0
765   m.id=0
766 -endif
767
768 set talk off
769
770 **OPEN FILE # 1
771
772 -if used("MCFEITMP")
773   select cfetmp
774   set order to tag hmcfeid of hmcfe.cdx
775 -else
776   select 0
777   use hmcfe alias cfetmp again
778   set order to tag hmcfeid of hmcfe.cdx
779 -endif
780
781 **OPEN FILE # 2
782
783 -if used("MCFETMP")
784   select cfetmp
785   set order to tag hmcfid of hmcfe.cdx
786 -endif
787
788
789
790
791

```

**WNTAB.AC1 10-28-92 1:47p**

Page 7 of 11



```

987 else
988   m.hmcfe=nn
989   m.olchmcfe=nn
990 endif
991
992 if m.hmcfeid>0
993   m.hmcfe=get_cfar(m.hmcfeid)
994   m.olchmcfe=m.hmcfe
995 else
996   m.hmcfe=nn
997   m.olchmcfe=nn
998 endif
999
1000 if m.hmcfeid>0
1001   m.hmcfe=get_cfar1(m.hmcfeid)
1002
1003 else
1004   m.hmcfe=nn
1005
1006 endif
1007
1008 if m.hmcfeid>0
1009   m.hmcfe=get_el(m.hmcfeid)
1010
1011 else
1012   m.hmcfe=nn
1013
1014 endif
1015
1016 show gets
1017 show get action enabled
1018 show get m.element disabled
1019 show get save disabled
1020 show get m.halc disabled
1021 show get m.hmap disabled
1022 show get m.hmc disabled
1023 show get m.hmcprob disabled
1024 show get m.hmcfe disabled
1025 show get m.hmcfeconst disabled
1026 show get m.prob disabled
1027 * SHOW GET m.hmunit DISABLED
1028 show get m.perp disabled
1029 show get m.perd disabled
1030 show get m.perq disabled
1031
endif

```

_*OXOLYR5Y	m.halc WHEN	
Function Origin:		
From Screen:	HTAB,	Record Number: 8
Variable:	m.halc	
Called By:	WHEN Clause	
Object Type:	Field	
Snippet Number:	2	

```

1053 *
1054 *
1055 *
1056 *
1057 *
1058 *
1059 *
1060 *
1061 *
1062 *
1063 *
1064 *
1065 *
1066 *
1067 *
1068 *
1069 *
1070 *
1071 *
1072 *
1073 *
1074 *
1075 *
1076 *
1077 *
1078 *
1079 *
1080 *
1081 *
1082 *
1083 *
1084 *
1085 *
1086 *
1087 *
1088 *
1089 *
1090 *
1091 *
1092 *
1093 *
1094 *
1095 *
1096 *
1097 *
1098 *
1099 *
1100 *
1101 *
1102 *
1103 *
1104 *
1105 *
1106 *
1107 *
1108 *
1109 *
1110 *
1111 *
1112 *
1113 *
1114 *
1115 *
1116 *
1117 *
1118 *

```

**\_Q9XOLYR8Z**      **m.hmlc VALID**

Function Origin:                      HMTAB,      Record Number:    8

From Screen:                      m.hmlc

Variable:                      VALID Clause

Called By:                      Field

Object Type:                      3

Snippet Number:                      3

```

1119 *
1120 *
1121 *
1122 *
1123 *
1124 *
1125 *
1126 *
1127 *
1128 *
1129 *
1130 *
1131 *
1132 *
1133 *
1134 *
1135 *
1136 *
1137 *
1138 *
1139 *
1140 *
1141 *
1142 *
1143 *
1144 *
1145 *
1146 *
1147 *
1148 *
1149 *
1150 *
1151 *
1152 *
1153 *
1154 *
1155 *
1156 *
1157 *
1158 *
1159 *
1160 *
1161 *
1162 *
1163 *
1164 *
1165 *
1166 *
1167 *
1168 *
1169 *
1170 *
1171 *
1172 *
1173 *
1174 *
1175 *
1176 *
1177 *
1178 *
1179 *
1180 *
1181 *
1182 *
1183 *
1184 *

```

```

function _q9x0lyrbz    && m.hmlc VALID
#region 1
if m.oldhmlc <> m.hmlc
  m.hmlc=trim(m.hmlc)
  m.hmlc=if(m.hmlc="?", "", upper(m.hmlc))
  m.hmlc=get_hmlc(m.hmlc)
  if not empty(m.hmlc)
    select distinct hmlc.hmlcid;
    from hmlc;
    where hmlc.hmlc in (m.hmlc);
    into array x
    m.hmlcid=x[1]
  endif
  show gets
endif

```

**\_Q9XOLYR08**      **m.hmap WHEN**

Function Origin:                      HMTAB,      Record Number:    10

From Screen:                      m.hmap

Variable:                      WHEN Clause

Called By:                      Field

Object Type:                      4

Snippet Number:                      4

```

1185 *
1186 *
1187 *
1188 *
1189 *
1190 *
1191 *
1192 *
1193 *
1194 *
1195 *
1196 *
1197 *
1198 *
1199 *
1200 *
1201 *
1202 *
1203 *
1204 *
1205 *
1206 *
1207 *
1208 *
1209 *
1210 *
1211 *
1212 *
1213 *
1214 *
1215 *
1216 *
1217 *
1218 *
1219 *
1220 *
1221 *
1222 *
1223 *
1224 *
1225 *
1226 *
1227 *
1228 *
1229 *
1230 *
1231 *
1232 *
1233 *
1234 *
1235 *
1236 *
1237 *
1238 *
1239 *
1240 *
1241 *
1242 *
1243 *
1244 *
1245 *
1246 *
1247 *
1248 *
1249 *
1250 *
1251 *
1252 *
1253 *
1254 *
1255 *
1256 *
1257 *
1258 *
1259 *
1260 *
1261 *
1262 *
1263 *
1264 *
1265 *
1266 *
1267 *
1268 *
1269 *
1270 *
1271 *
1272 *
1273 *
1274 *
1275 *
1276 *
1277 *
1278 *
1279 *
1280 *
1281 *
1282 *
1283 *
1284 *

```

```

function _q9x0lyr08    && m.hmap WHEN
#region 1
m.oldhmap=m.hmap

```

**\_Q9XOLYR06**      **m.hmap VALID**

Function Origin:                      HMTAB,      Record Number:    10

From Screen:                      m.hmap

Variable:                      VALID Clause

Called By:                      Field

Object Type:                      5

Snippet Number:                      5

```

1285 *
1286 *
1287 *
1288 *
1289 *
1290 *
1291 *
1292 *
1293 *
1294 *
1295 *
1296 *
1297 *
1298 *
1299 *
1300 *
1301 *
1302 *
1303 *
1304 *
1305 *
1306 *
1307 *
1308 *
1309 *
1310 *
1311 *
1312 *
1313 *
1314 *
1315 *
1316 *
1317 *
1318 *
1319 *
1320 *
1321 *
1322 *
1323 *
1324 *
1325 *
1326 *
1327 *
1328 *
1329 *
1330 *
1331 *
1332 *
1333 *
1334 *
1335 *
1336 *
1337 *
1338 *
1339 *
1340 *
1341 *
1342 *
1343 *
1344 *
1345 *
1346 *
1347 *
1348 *
1349 *
1350 *
1351 *
1352 *
1353 *
1354 *
1355 *
1356 *
1357 *
1358 *
1359 *
1360 *
1361 *
1362 *
1363 *
1364 *
1365 *
1366 *
1367 *
1368 *
1369 *
1370 *
1371 *
1372 *
1373 *
1374 *
1375 *
1376 *
1377 *
1378 *
1379 *
1380 *
1381 *
1382 *
1383 *
1384 *

```

```

function _q9x0lyru6    && m.hmap VALID
#region 1
if m.oldhmap <> m.hmap
  m.hmap=trim(m.hmap)
  m.hmap=if(m.hmap="?", "", upper(m.hmap))
  m.hmap=get_hmap(m.hmap)
  if not empty(m.hmap)
    select distinct hmap.hmapid;
    from hmap;
    where hmap.hmap in (m.hmap);
    into array x
    m.hmapid=x[1]
  endif
  show gets
endif

```



```

1317 if m.hmlcid>0
1318   m.hmlc=get_hmlcn(m.hmlcid)
1319   m.oldhmlc=m.hmlc
1320 else
1321   m.hmlc=""
1322   m.oldhmlc=""
1323 endif
1324
1325 if m.hmapid>0
1326   m.hmap=get_hmapn(m.hmapid)
1327   m.oldhmap=m.hmap
1328 else
1329   m.hmap=""
1330   m.oldhmap=""
1331 endif
1332
1333 if m.hmetid>0
1334   m.hmet=get_hmetn(m.hmetid)
1335   m.oldhmet=m.hmet
1336 else
1337   m.hmet=""
1338   m.oldhmet=""
1339 endif
1340
1341 if m.hmcfid>0
1342   m.hmcfc=get_cfar(m.hmcfid)
1343   m.oldhmcfc=m.hmcfc
1344 else
1345   m.hmcfc=""
1346   m.oldhmcfc=""
1347 endif
1348
1349 if m.hmcfeid>0
1350   m.hmcfe=get_cfar1(m.hmcfeid)
1351 else
1352   m.hmcfe=""
1353 endif
1354
1355 if m.hmcfelid>0
1356   m.hmcfel=get_el(m.hmcfelid)
1357 else
1358   m.hmcfel=""
1359 endif
1360 show gets
1361 show get action enabled
1362 show get m.element disabled
1363 show get save disabled
1364 show get m.hmlc disabled
1365 show get m.hmap disabled
1366 show get m.hmet disabled
1367 show get m.hmcfc disabled
1368 show get m.hmcf disabled
1369 show get m.hmcfecost disabled
1370 show get m.prob disabled
1371 show get m.hunit disabled
1372 show get m.perp disabled
1373 show get m.perd disabled
1374 show get m.perq disabled
1375 m.adding=.f.
1376 m.change=.f.
1377 *: EOF: WNTAB.AC1
1378

```

[illegible]

## HMSTEP Screen Layout

```

133 *
134 *
135 *
136 *
137 *
138 #region 1
139 if visible("hmascen")
140   activate window hmascen same
141 else
142   activate window hmascen noahow
143 endif
144
145 a 2,2 say "Material:"
146 a 3,2 say "Phase:"
147 a 4,2 say "Process:"
148 a 7,6 say "Number of Employees:"
149 a 8,6 say "Quantity of Material:"
150 a 1,1 to 9,63
151 a 8,34 say "Unit:"
152 a 0,2 say "Step #:"
153 a 0,9 get m.hmstep ;
154   size 1,4 ;
155   default 0 ;
156   disable
157 a 0,54 get m.gmscid ;
158   size 1,4 ;
159   default 0 ;
160   picture "a2" ;
161   disable
162 a 0,60 get m.gmscname ;
163   size 1,15 ;
164   default " " ;
165   picture "a1" ;
166   disable
167 a 2,12 get m.hmatid ;
168   size 1,4 ;
169   default 0 ;
170   picture "a2" ;
171   disable
172 a 2,17 get m.hname ;
173   size 1,46 ;
174   default " " ;
175   picture "a1" ;
176   when _qbrdy3jhw() ;
177   valid _qbrdy3jhw() ;
178 a 2,65 get m.action
179   picture "a1" \<Add;\<Edit;\<Next;\<Previous;\<Quit" ;
180   size 1,10,1 ;
181   default 1 ;
182   valid _qbrdy3j3t() ;
183   disable
184 a 3,12 get m.halcid ;
185   size 1,4 ;
186   default 0 ;
187   picture "a2" ;
188   disable
189 a 3,17 get m.halc ;
190   size 1,46 ;
191   default " " ;
192   picture "a1" ;
193   when _qbrdy3j3q() ;
194   valid _qbrdy3j3q() ;
195 a 4,12 get m.hmapid ;
196   size 1,4 ;
197   default 0 ;
198   picture "a2" ;

```

```

199 disable
200 a 4,17 get m.hmap ;
201   size 2,46 ;
202   default " " ;
203   picture "a1" ;
204   when _qbrdy3j3k7() ;
205   valid _qbrdy3j3k7() ;
206 a 7,28 get m.pernum ;
207   size 1,3 ;
208   default 0 ;
209 a 7,50 get m.durnum ;
210   size 1,9 ;
211   default 0 ;
212 a 8,28 get m.qtnum ;
213   size 1,3 ;
214   default 0 ;
215 a 8,40 get m.unit ;
216   size 1,19 ;
217   default " " ;
218 a 11,27 get m.save ;
219   picture "a1" \<Save;\<Cancel" ;
220   size 1,8,1 ;
221   default 1 ;
222   valid _qbrdy3j13() ;
223 a 0,44 say "Scenario:"
224 a 7,34 say "Number of days:"
225
226 if not visible("hmascen")
227   activate window hmascen
228 endif
229
230 read cycle modal ;
231   when _qbrdy3j1qz()
232
233 release window hmascen
234 select (m.curarea)
235
236 #region 0
237 if m.talkstat = "ON"
238   set talk on
239 endif
240 if m.compstat = "ON"
241   set compatible on
242 endif
243
244 *
245 *
246 *
247 *
248 *
249 *
250 *
251
252 #region 1
253 pop key all
254 set escape &oldescape
255 set safety &oldsafe
256 return m.quitflag
257 ===== End of Main Body - Entry Cleanup
258 procedure expresse
259 return
260
261 *****
262 procedure initvar
263 *****
264

```

HMSTEP Cleanup Code

```

265 if m.hmatid=0
266   m.hname=get_hmatn(m.hmatid)
267   m.olchname=m.hname
268 else
269   m.hname=space(46)
270   m.olchname=space(46)
271 endif
272
273 if m.hmlcid=0
274   m.halc=get_halcn(m.hmlcid)
275   m.olchmlc=m.halc
276 else
277   m.halc=space(46)
278   m.olchmlc=space(46)
279 endif
280
281 if m.hmapid=0
282   m.hmap=get_hmapn(m.hmapid)
283   m.olchmap=m.hmap
284 else
285   m.hmap=space(80)
286   m.olchmap=space(80)
287 endif
288 return
289 *****
290 procedure adoption
291 *****
292 show get m.action disabled
293 show get m.save enabled
294 show get m.hname enabled
295 show get m.halc enabled
296 show get m.hmap enabled
297 show get m.pernum enabled
298 show get m.qtrnum enabled
299 show get m.unit enabled
300 return
301
302 *****
303 procedure change
304 *****
305 m.olchact = set( "EXACT" )
306 set exact on
307 m.change =(htemp.hmatid <> m.hmatid;
308 or htemp.hmlcid <> m.hmlcid;
309 or htemp.hmapid <> m.hmapid;
310 or htemp.qtrnum <> m.qtrnum;
311 or htemp.pernum <> m.pernum;
312 or upper(trim(htemp.unit)) <> upper(trim(m.unit)))
313
314 set exact &oldact
315 return m.change
316
317 *****
318 function get_hmat
319 *****
320 parameter name
321 release hmat
322 dimension hmat[1]
323 hmat[1]=" "
324 m.olchfile=select(
325   select hmat.hname, hmat.hmatid;
326   from hmat;
327   where hmat.hname in (alltrim(m.name));
328
329
330

```

MASTER.ACT 10-28-92 1:48p

```

331 into array hmat
332 m.ans=""
333 if not empty(hmat[1])
334   m.ans=chooser(hmat,"Select a Material")
335   if empty(m.ans)
336     m.index = ascn(hmat,m.ans)
337     m.id = hmat[m.index + 1]
338     m.ans = m.ans + u + str(m.id)
339   endif
340 else
341   zerrmsg(m.name + " was not found",1)
342   m.ans=""
343   endif
344
345 select (m.olchfile)
346 return m.ans
347
348 *****
349 function get_hmatn
350 *****
351 parameter m.id
352
353 select hmat.hmatname;
354   from hmat;
355   where hmat.hmatid = m.id;
356   into array x
357
358 m.name=x[1]
359 return m.name
360
361 *****
362 *GET LIFE CYCLE PHASE
363 function get_hmlc
364 *****
365 parameter m.hmlc
366
367 release hmlc
368 dimension hmlc[1]
369 hmlc[1]=" "
370 m.olchfile=select(
371   select hmlc.hmlc, hmlc.hmlcid;
372   from hmlc;
373   where hmlc.hmlc in (alltrim(m.hmlc));
374   order by hmlcid;
375   into array hmlc
376
377 m.ans=""
378 if not empty(hmlc[1])
379   m.ans=chooser(hmlc,"Select a Life Cycle Phase")
380   if empty(m.ans)
381     m.index = ascn(hmlc,m.ans)
382     m.id = hmlc[m.index + 1]
383     m.ans = m.ans + u + str(m.id)
384   endif
385 else
386   zerrmsg(m.hmlc + " was not found",1)
387   endif
388
389 select (m.olchfile)
390 return m.ans
391
392 *****
393 function get_hmlcn
394 *****
395 parameter m.hmlc
396

```

Page 3 of 7

```

397 parameter m.id
398
399 select hmlc,hmlc;
400 from hmlc;
401 where hmlc.hmlcid = m.id;
402 into array x
403
404 m.name=x[1]
405 return m.name
406
407 *****
408 *Working Processes
409 function get_hmap
410 *****
411 parameter m.match
412 release hmap
413 dimension hmap[1]
414 hmap[1]=""
415 m.oldfile=select()
416 m.match = upper(trim(m.match))
417 select hmap,hmap,hmap.hmapid;
418 from hmap;
419 where upper(hmap.hmap) in (alltrim(m.match));
420 into array hmap
421
422 m.ans=""
423 if not empty(hmap[1])
424 m.ans=chooser(hmap,"Select a Process")
425
426 if empty(m.ans)
427 m.id = ascscan(hmap,m.ans)
428 m.id = hmap[m.idex + 1]
429 m.ans = m.ans + u + str(m.id)
430
431 else
432 m.ans=m.hmap + " was not found",1)
433 endif
434
435 select (m.oldfile)
436
437 return m.ans
438
439 *****
440 function get_hmapn
441 *****
442 parameter m.id
443
444 select hmap,hmap;
445 from hmap;
446 where hmap.hmapid = m.id;
447 into array x
448
449 m.name=x[1]
450 return m.name
451
452 *****
453 function get_hmapn
454 *****
455 private m.unit
456 m.unit = ""
457 if empty(m.hmapid) and empty(m.hmlcid) and empty(m.hmapid)
458 dimension x[1]
459 x[1]=""
460 select hmlc,hmlc,hmlc.hmapid;
461 from hmlc;
462 where hmlc.hmlcid = m.hmlcid;

```

```

463 and hmlc.hmlcid = m.hmlcid;
464 and hmlc.hmapid = m.hmapid;
465 into array x
466
467 if empty(x[1])
468 for i = 1 to alen(x) step 2
469 m.unit = x[i+1]
470 if empty(m.unit)
471 exit
472 endif
473 m.validproc = .t.
474 show gets
475
476 else
477 m.ans="This Material Processing isn't defined in the table"
478 m.validproc = .f.
479 endif
480 return m.unit
481
482 *****
483 procedure scsave
484 *****
485 msel = select()
486 select hmap
487 if schange
488 set order to tag hmlcid
489 delete for hmlcid = m.hmlcid
490 pack
491 append from hmap
492 use hmlc
493 set order to tag hmlcid
494 if lseek(m.hmlcid)
495 append blank
496 replace hmlcid with m.hmlcid
497 replace hmlcname with m.hmlcname
498 endif
499
500 *****
501 function datacheck
502 *****
503
504 do case
505 case empty(m.hmlcid)
506 = errmsg("Missing Material",1)
507 curobj = objnum(hmlc)
508 check = .f.
509
510 case empty(m.hmlcid)
511 = errmsg("Missing Life Cycle Phase",1)
512 curobj = objnum(hmlc)
513 check = .f.
514
515 case empty(m.hmapid)
516 = errmsg("Missing Working Process",1)
517 curobj = objnum(hmap)
518 check = .f.
519
520 case empty(m.pernum) and empty(m.durnum) and empty(m.qtynum)
521 = errmsg("Enter either number of People, Duration, or Quantity",1)
522 curobj = objnum(pernum)
523 check = .f.
524
525 otherwise
526 =get_hmapn()
527 if !m.validproc
528 curobj = objnum(hmlc)
529 check = .f.
530
531 else
532 check = .t.
533 endif
534 endcase

```



```

661 *
662 *
663 *
664 *
665 *
666 *
667 *
668 *
669 *
670 *
671 *
672 *
673 *
674 *
675 *
676 *
677 *
678 *
679 *
680 *
681 *
682 *
683 *
684 *
685 *
686 *
687 *
688 *
689 *
690 *
691 *
692 *
693 *
694 *
695 *
696 *
697 *
698 *
699 *
700 *
701 *
702 *
703 *
704 *
705 *
706 *
707 *
708 *
709 *
710 *
711 *
712 *
713 *
714 *
715 *
716 *
717 *
718 *
719 *
720 *
721 *
722 *
723 *
724 *
725 *
726 *

```

```

_Q8R0Y3JSG      m.hmlc WHEN
Function Origin:
From Screen:      HMSTEP,      Record Number: 17
Variable:          m.hmlc
Called By:         WHEN Clause
Object Type:       Field
Snippet Number:    4

```

```

function _q8r0y3jsg      && m.hmlc WHEN
#region 1
m.oldhmlc=m.hmlc

```

```

_Q8R0Y3JYQ      m.hmlc VALID
Function Origin:
From Screen:      HMSTEP,      Record Number: 17
Variable:          m.hmlc
Called By:         VALID Clause
Object Type:       Field
Snippet Number:    5

```

```

function _q8r0y3jq      && m.hmlc VALID
#region 1
if m.oldhmlc <> m.hmlc
  m.hmlc=trim(m.hmlc)
  m.hmlc=if(m.hmlc="?", "", upper(m.hmlc))
  m.hmdate=get_hmlc(m.hmlc)
  if not empty(m.hmdate)
    m.hmlc = dp(m.hmdate,u,1)
    m.hmlcid = val(dp(m.hmdate,u,2))
  else
    m.hmlc = space(46)
  endif
  show gets
  m.t1 = get_hmlc()
  if empty(m.t1)
    m.unit = m.t1
  endif
endif
show gets
endif

```

```

_Q8R0Y3KC7      m.hmlc WHEN
Function Origin:
From Screen:      HMSTEP,      Record Number: 19
Variable:          m.hmlc
Called By:         WHEN Clause
Object Type:       Field
Snippet Number:    6

```

```

727 *
728 *
729 *
730 *
731 *
732 *
733 *
734 *
735 *
736 *
737 *
738 *
739 *
740 *
741 *
742 *
743 *
744 *
745 *
746 *
747 *
748 *
749 *
750 *
751 *
752 *
753 *
754 *
755 *
756 *
757 *
758 *
759 *
760 *
761 *
762 *
763 *
764 *
765 *
766 *
767 *
768 *
769 *
770 *
771 *
772 *
773 *
774 *
775 *
776 *
777 *
778 *
779 *
780 *
781 *
782 *
783 *
784 *
785 *
786 *
787 *
788 *
789 *
790 *
791 *
792 *

```

```

_Q8R0Y3KI1H      m.hmlc VALID
Function Origin:
From Screen:      HMSTEP,      Record Number: 19
Variable:          m.hmlc
Called By:         VALID Clause
Object Type:       Field
Snippet Number:    7

```

```

function _q8r0y3kih      && m.hmlc VALID
#region 1
if m.oldhmlc <> m.hmlc
  m.hmlc=trim(m.hmlc)
  m.hmlc=if(m.hmlc="?", "", upper(m.hmlc))
  m.hmdate=trim(get_hmlc(m.hmlc))
  if not empty(m.hmdate)
    m.hmlc = dp(m.hmdate,u,1)
    m.hmlcid = val(dp(m.hmdate,u,2))
  else
    m.hmlc = space(80)
  endif
  show gets
  m.t1 = get_hmlc()
  if empty(m.t1)
    m.unit = m.t1
  endif
endif
show gets
endif

```

```

_Q8R0Y3L13      m.Save VALID
Function Origin:
From Screen:      HMSTEP,      Record Number: 24
Variable:          m.Save
Called By:         VALID Clause
Object Type:       Push Button
Snippet Number:    8

```

```

function _q8r0y3l13      && m.Save VALID
#region 1
do case
case m.save = 1 && Selected Save Button
  m.check = datacheck()
  if m.check
    m.hmlcid = m.ghmlcid
  if m.adding && Adding a new record

```

return check

```

529 *
530 *
531 *
532 *
533 *
534 *
535 *
536 *
537 *
538 *
539 *
540 *
541 *
542 *
543 *
544 *
545 *
546 *
547 *
548 *
549 *
550 *
551 *
552 *
553 *
554 *
555 *
556 *
557 *
558 *
559 *
560 *
561 *
562 *
563 *
564 *
565 *
566 *
567 *
568 *
569 *
570 *
571 *
572 *
573 *
574 *
575 *
576 *
577 *
578 *
579 *
580 *
581 *
582 *
583 *
584 *
585 *
586 *
587 *
588 *
589 *
590 *
591 *
592 *
593 *
594 *

```

\_Q8R0Y3IHU

Function Origin:

m.hname WHEN

From Screen:

HMSTEP,

m.hname

WHEN Clause

Field

1

Record Number:

14

Object Type:

Snippet Number:

1

function q8r0y3ihw

#region 1

m.oldhname = m.hname

if m.hname

then

m.hname = m.hname

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

\_Q8R0Y3IQX

Function Origin:

m.hname VALID

From Screen:

HMSTEP,

m.hname

VALID Clause

Field

2

Record Number:

14

Object Type:

Snippet Number:

2

function q8r0y3iqx

#region 1

if m.hname

then

m.hname = m.hname

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

endif

Object Type: Push Button

Snippet Number: 3

function q8r0y3jst

#region 1

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

m.hname = m.hname

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

```

793 append blank
794 gather memvar
795 m.change = .t.
796
797 else
798 do change
799 if m.change && Changing an old record
800 gather memvar
801 m.change = .t.
802 endif
803 m.oldrec = recno()
804
805 else
806 return
807 endif
808 case m.save = 2
809 if recount() = 0
810 quit = .f.
811 if empty(m.hname) and empty(m.hmlc) and empty(hmap)
812 quit = yesno("Do you want to", "Quit", "Continue")
813 endif
814 if quit
815 m.quitflag = .t.
816 clear read
817 else
818 scatter memvar blank
819 do initvar
820 show gets
821 do adoption
822 m.adding = .t.
823 m.change = .f.
824 m.quitflag = .f.
825 return
826 endif
827 endcase
828 if empty(m.oldrec)
829 go m.oldrec
830 scatter memvar
831 endif
832 do initvar
833 show gets disabled
834 show get action enabled
835
836 m.adding=.f.
837 m.change=.f.
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853 function _q8r0y3lqz && Read Level When
854
855 * When Code from screen: HMSTEP
856
857 $region 1
858 if m.adding

```

_q8r0y3lqz	Read Level When
Function Origin:	
From Screen:	HMSTEP
Called By:	READ Statement
Snippet Number:	9

```

859 m.action = 1
860 show gets disable
861 show get action enabled
862 endif
864 *: EOF: HMSTEP.ACT

```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

```

if set("TALK") = "ON"
  set talk off
  m.talkstat = "ON"
else
  m.talkstat = "OFF"
endif
m.compat = set("COMPATIBLE")
set compatible foxplus
m.currecs = select()

```

```

if not wexist("w_hmcent")
define window w_hmcent :
from int((srow()-7)/2),int((scol()-66)/2) :
to int((srow()-7)/2)+6,int((scol()-66)/2)+65 :
title "Cost Analysis" :
nfloat :
noclose :
shadow :
double :
color scheme 1
endif

```

```

region 1
n.hmscen = ""
n.hmscenid = 0
n.driven = ""
sevent = 1
mstatus = 1

```

**HMSCEN Cleanup Code**

```

133 use
134 select (me1)
135 m.ans = m.driven + u + alltrim(str(m.hmscenid)) + u + m.hmscen
136 return m.ans
137
138 *-----
139 * This procedure will return:
140 * Scenario name (Characters) Scenario id (Numeric)
141 *-----
142 function get_hmsc
143 parameter name
144 release hmscn
145 dimension hmscn(1)
146
147 name = alltrim(name)
148 select distinct hmscen.hmscname, hmscen.hmscid;
149 from hmscen;
150 where upper(hmscen.hmscname) in (upper(name));
151 into array hmscn
152
153 m.ans=""
154 if not empty(hmscn(1))
155   m.ans=chooser(hmscn,"Select a Material Scenario")
156   if empty(m.ans)
157     m.index=ascn(hmscn,m.ans)
158     m.id = hmscn[m.index + 1]
159     m.ans = alltrim(m.ans) + u + alltrim(str(m.id))
160   endif
161   return m.ans
162 endif
163
164 *-----
165 * This procedure show the status of option.
166 * status 1: if new entry allow for add, browse, and cancel
167 * status 2: if entry already exist allow for retrieve, delete, browse
168
169
170
171
172 procedure mevent
173 parameter mevent
174 if new
175   show get mevent,1 enabled
176   show get mevent,2 disabled
177   show get mevent,3 disabled
178 else
179   show get mevent,1 disabled
180   show get mevent,2 enabled
181   show get mevent,3 enabled
182 endif
183
184 return
185
186
187
188
189
190
191
192
193
194
195
196
197

```

\_QBK0INGJH

Function Origin:

From Screen:

Variable:

Called By:

Object Type:

Snippet Number:

Record Number: 3

HMSCEN

m.hmscen

VALID Clause

Field 1

```

198
199
200 function _qbk0ingjh      ** m.hmscen VALID
201 #region 1
202 m.hmsc = ""
203 if not empty(m.hmscen)
204   m.hmscen=alltrim(m.hmscen)
205   m.hmsc=if(m.hmscen=""**,**,upper(m.hmscen))
206   m.hmsc=get_hmsc(m.hmscen)
207 endif
208 if empty(m.hmsc)
209   do mevent with .t.
210 else
211   do mevent with .f.
212   m.hmscen = dp(m.hmsc, u, 1)
213   m.hmscenid = val(dp(m.hmsc, u, 2))
214 endif
215 show gets
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231 function _qbk0ingtv      ** mevent VALID
232 #region 1
233 if empty(m.hmscen)
234   curobj = objnum(m.hmscen)
235   zerrmsg("Input required",2)
236   show gets
237   return
238 endif
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260

```

\_QBK0INGTV

Function Origin:

From Screen:

Variable:

Called By:

Object Type:

Snippet Number:

mevent VALID

HMSCEN

mevent

VALID Clause

Push Button

2

\*\* Brow

\*\* Dele

\*\* Retr

\*\* New

```

261 m.hnsc = get_hnsc(m.hnsc)
262 if isempty(m.hnsc)
263     do event with .f.
264         m.hnscen = dp(m.hnsc,u,1)
265         m.hnscenid = val(dp(m.hnsc,u,2))
266     end
267     show gets
268
269     case event = 5
270         m.hnscenid = 0
271         m.hnscen = ""
272         m.driven = "C"
273         clear read
274     endcase
275     return
276
277 *
278 *
279 *
280 *
281 *
282 *
283 *
284 *
285 *
286 *
287 *
288 *
289 *
290 *
291 *
292 *
293 *
294 *
295 *
296
function _qtk01nhbv    % Read Level when
% When Code from screen: HNMCEN
%
%region 1
do event with .t.
%: EOF: HNMCEN.ACT

```

_qtk01nhbv	Read Level when
Function Origin:	HNMCEN
From Screen:	READ Statement
Called By:	3
Snippet Number:	

```

1 *****
2 Procedure file: C:\MILCOM\WORK\HMSC.PRG
3
4 System: Hazardous Material Life-Cycle Cost
5 Author: Ly, Hoa
6 Copyright (c) SEPT 1992, Naval Health Research Center
7 Last modified: 10/20/92
8
9 Procs & Fncts: CLOSEFILE
10 : DEL HMSC
11 : FILEFIND
12 : OPENFILE
13 : POPUPSHOW
14 : POPUPHIDE
15
16 Set by: HMEU.NPR
17
18 Calls: SET()
19 : HMSCN()
20 : DP()
21 : INIT()
22 : VAL()
23 : EMPTY()
24 : HMSCP()
25 : DEL HMSC
26 : YESNO()
27 : HMSCOMP.SPR
28 : CLOSEFILE
29
30 (function in ?)
31 (function in HMSCN.PRG)
32 (function in DP.PRG)
33 (function in ?)
34 (function in ?)
35 (function in ?)
36 (function in HMSCP.PRG)
37 (procedure in HMSC.PRG)
38 (function in YESNO.PRG)
39 (procedure in HMSC.PRG)
40
41 Documented 10/28/92 at 13:47 FoxDoc version 2.10
42 *****
43 public mtalk, mesc, u
44 if set("TALK") = "ON"
45 set talk off
46 mtalk = "ON"
47 endif
48 mesc = set("ESCAPE")
49 set escape off
50 u = ""
51 m.quitflag = .f.
52 mdata = hmscnc()
53 mevent = dp(mdata,u,1)
54 m.hmscid = int(val(dp(mdata,u,2)))
55 m.hmscname = dp(mdata,u,3)
56
57 do case
58 case mevent = "C" or empty(mevent)
59 return
60 case mevent = "M"
61 m.adding = .t.
62 m.quitflag = hmscp(m.adding, m.hmscid, m.hmscname)
63 case mevent = "H"
64 m.adding = .f.
65 do hmscp with m.adding, m.hmscid, m.hmscname
66 m.quitflag = .f.
67 case mevent = "D"
68 m.hmscid = del_hmsc(m.hmscid)
69 m.quitflag = .t.
70 otherwise
71 return
72 endcase
73 if m.quitflag
74 m.compute = yesno("Do Computation Now?", "Yes", "No")
75 if m.compute
76 do hmscomp.spr with m.hmscid, m.hmscname
77
78 HMSC.ACT 10-28-92 1:47p

```

```

67 Lendif
68 do closefile
69
70 return
71
72 *****
73 Procedure: CLOSEFILE
74 Called by: HMSC.PRG
75
76 Calls: EMPTY()
77 (function in ?)
78
79 *****
80 procedure closefile
81 *****
82 close database
83 if empty(mesc)
84 set escape &mesc
85 endif
86 if empty(mtalk)
87 set talk &mtalk
88 endif
89 return
90
91 *****
92 Procedure: DEL_HMSC
93 Called by: HMSC.PRG
94
95 Calls: POPUPSHOW
96 : OPENFILE
97 : POPUPHIDE
98 (procedure in HMSC.PRG)
99 (procedure in HMSC.PRG)
100 (procedure in HMSC.PRG)
101
102 *****
103 function del_hmsc
104 *****
105 parameter id
106 wpop = popupshow("Deleting...")
107 filesuccess = openfile("hmscp")
108 set order to tag hmscid
109 if filesuccess
110 delete for hmscid = m.id
111 pack
112 endif
113 wpophide(wpop)
114 return
115
116 *****
117 * Find the location of the file and file path will update into new pa
118 * if the file path has not in the foxpro path yet.
119
120 > th
121
122
123
124
125 *****
126 Procedure: FILEFIND
127 Called by: OPENFILE
128 (procedure in HMSC.PRG)
129
130 Calls: PARAMETER()
131 (function in ?)
132
133 *****

```

```

144 *1 : FILE() (function in ?)
145 *1 : DP() (function in DP.PRG)
146 *1 : LOCFILE() (function in ?)
147 *1
148 *****
149 function filefind
150 *****
151 parameter mfilename
152 private mflag
153 mflag = .t.
154 if parameter() = 0
155 return .f.
156 endif
157 on error return .f.
158 if ifile(mfilename)
159 mfile = dp(mfilename, " ", 1)
160 mext = dp(mfilename, " ", 2)
161 -locfile(mfile, mext, "where is the " + mfilename + "?.")
162 endif
163 return mflag
164 *****
165 *****
166 *****
167 *****
168 *****
169 *****
170 *****
171 *****
172 *****
173 *****
174 *****
175 *****
176 *****
177 *****
178 *****
179 *****
180 *****
181 *****
182 *****
183 *****
184 *****
185 *****
186 *****
187 *****
188 *****
189 *****
190 *****
191 *****
192 *****
193 *****
194 *****
195 *****
196 *****
197 *****
198 *****
199 *****
200 *****
201 *****
202 *****
203 *****
204 *****
205 *****
206 *****
207 *****
208 *****
209 *****
210 *****
211 *****

```

```

212 *1 : SCOL() (function in ?)
213 *1 : WVISIBLE() (function in ?)
214 *1
215 *****
216 function popshow
217 *****
218 parameters errstr
219 *****
220 if not mexist("u_popnote")
221 define window u_popnote
222 from int((srow()-8)/2), int((scol()-36)/2) ;
223 to int((srow()-8)/2)+7, int((scol()-36)/2)+35 ;
224 title "One moment" ;
225 float ;
226 close ;
227 shadow ;
228 double ;
229 color scheme 5
230 endif
231 if visible("u_popnote")
232 activate window u_popnote same
233 else
234 activate window u_popnote noshow
235 endif
236 a 1,1 say errstr size 3,31
237
238 if not visible("u_popnote")
239 activate window u_popnote
240 endif
241 return ""
242 *****
243 *****
244 *****
245 *****
246 *****
247 *****
248 *****
249 *****
250 *****
251 *****
252 *****
253 *****
254 *****
255 *****
256 *****
257 *****
258 *****
259 *****
260 *****
261 *****
262 *****
263 *****
264 *****
265 *****
266 *****
267 *****
268 *****
269 *****
270 *****
271 *****
272 *****
273 *****
274 *****
275 *****
276 *****
277 *****
278 *****
279 *****
280 *****
281 *****
282 *****
283 *****
284 *****
285 *****
286 *****
287 *****
288 *****
289 *****
290 *****
291 *****
292 *****
293 *****
294 *****
295 *****
296 *****
297 *****
298 *****
299 *****
300 *****
301 *****
302 *****
303 *****
304 *****
305 *****
306 *****
307 *****
308 *****
309 *****
310 *****
311 *****

```



```

1 *****
2
3 Procedure file: C:\HMLCCM\WORK\HMINIT.PRG
4
5 System: Hazardous Material Life-Cycle Cost
6 Author: LY, Noe
7 Copyright (c) SEPT 1992, Naval Health Research Center
8 Last modified: 09/09/92 8:49
9
10 Proc & Fncts: MYHANDLER()
11 : _QUIT
12
13 Calls: SET() (function in ?)
14 : UPPER() (function in ?)
15 : HMGTPUD() (function in HMGTPUD.PRG)
16 : IIF() (function in ?)
17 : BIGCHARS() (function in BIGCHARS.PRG)
18 : HMENU.MPR
19 : MYHANDLER() (function in HMINIT.PRG)
20
21 Documented 10/28/92 at 13:47 FoxDoc version 2.10
22 *****
23 *****
24
25 FUNCTION: HMINIT.PRG
26 PURPOSE: DRIVER FOR HMENU
27 NOTE: 3 CHANCES FOR ACCESSING ALLOWED.
28 PROCEDURES CALLED:
29 SIDE EFFECTS: NONE KNOWN.
30 REFERENCE: C:\EXAMPLE\PRGS\EX2.PRG
31 CREATED: 06/12/92 AL
32 MODIFIED: 06/14/92 AL
33
34 *****
35
36 *****
37
38 *****
39
40 Initialize system Init_app()
41
42 *****
43
44 close all
45 clear windows all
46 clear all
47 public m.pass,savetalk,savesc && Initialize public variables
48 public dropped
49
50 If set('TALK') = 'ON' && TALK handled as a special case.
51 set talk off && Turn TALK OFF
52 savetalk = 'ON' && TALK was ON, save the setting
53 else && TALK is OFF
54 savetalk = 'OFF' && TALK was OFF, save the setting
55 endif
56
57 If set('ESCAPE') = 'ON' && ESCAPE handled as a special case.
58 set escape off && Turn ESCAPE OFF
59 savesc = 'ON' && ESCAPE was ON, save the setting
60 else && ESCAPE is OFF
61 savesc = 'OFF' && ESCAPE was OFF, save the setting
62 endif
63
64 "HIDE WINDOW Command
65

```

```

63
64 *SET HELP TO HMINIT.DBF && Change help database
65 *SET HELP TO && TO DEFAULT HELP
66 set sysmenu to default && DEFAULT SYSTEM MENU
67
68 && End of Init_app()
69 *****
70
71 *
72 * Get password and print to screen*
73 *
74 *****
75 m.pass=upper(hmgetpud("")) && Get password
76
77 if m.pass != "SYSHAZ"
78 m.pass="HAZMAT"
79 endif
80
81 m.appbr=iif(m.pass="SYSHAZ","HMLCCM","HMLCCM") && Set title of appli
82 cation
83
84 =bigchars(6,17,m.appbr,6) && TITLE OF APPLICATION
85
86 dropped = .f.
87 push menu _msysmenu && HLL
88 do hmenu.mpr && Launch application menu
89 read valid myhandler()
90 pop menu _msysmenu && HLL
91 *****
92
93 * DONE
94 *****
95
96 clear windows all
97 close databases
98 close all
99 return
100 *****
101
102 * Function: MYHANDLER()
103 *
104 * Called by: APPLIC.PRG
105 *
106 *****
107
108 Function: MYHANDLER()
109 Called by: HMINIT.PRG
110 Calls: TYPE() (function in ?)
111
112 *****
113
114 function myhandler
115 if type("dropped") = "U"
116 dropped = .f.
117 endif
118 return dropped
119 *****
120
121 *****
122
123 *****
124
125 *****
126
127 *****
128
129 Procedure: _QUIT

```

```

130 *  

131 *      Called by: INMEMU.NPR  

132 *  

133 *-----  

127 function _quit  

128 dropped = .t.  

129 clear read all  

130 return  

131 *: EOF: ININIT.ACT  

132

```

```

1  *
2  *
3  *
4  *
5  *
6  *
7  *
8  *
9  *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *

```

09/09/92	HWLC.SPR	09:14:54
Dianna M. Pearsall & Hoa L. Ly Copyright (c) 1992 Naval Health Research Center Code 22 P.O.Box 85122 San Diego, CA 92186-5122 Description: This program was automatically generated by GENSCRN.		

```

67 *
68 *
69 *
70 *
71 *
72 *
73 *
74 *
75 *
76 *
77 *
78 *
79 *
80 *
81 *
82 *
83 *
84 *
85 *
86 *
87 *
88 *
89 *
90 *
91 *
92 *
93 *
94 *
95 *
96 *
97 *
98 *
99 *
100 *
101 *
102 *
103 *
104 *
105 *
106 *
107 *
108 *
109 *
110 *
111 *
112 *
113 *
114 *
115 *
116 *
117 *
118 *
119 *
120 *
121 *
122 *
123 *
124 *
125 *
126 *
127 *
128 *
129 *
130 *
131 *
132 *
133 *
134 *
135 *
136 *

```

```

region 0
regional m.curraea, m.talkstat, m.compstat

if set("TALK") = "ON"
  set talk off
  m.talkstat = "ON"
else
  m.talkstat = "OFF"
endif

m.compstat = set("COMPATIBLE")
set compatible foxplus

m.curraea = select()

```

S0795208 Databases, Indexes, Relations

```

100 *
101 *
102 *
103 *
104 *
105 *
106 *
107 *
108 *
109 *
110 *
111 *
112 *
113 *
114 *
115 *
116 *
117 *
118 *
119 *
120 *
121 *
122 *
123 *
124 *
125 *
126 *
127 *
128 *
129 *
130 *
131 *
132 *
133 *
134 *
135 *
136 *

```

```

if used("hwlc")
  select hwlc
  set order to 0
else
  select 0
  use (locfile("hwlc.dbf"), "DBF", "where is hwlc?");
  again alias hwlc ;
  order 0
endif

```

Window definitions

```

136 *
137 *
138 *
139 *
140 *
141 *
142 *
143 *
144 *
145 *
146 *
147 *
148 *
149 *
150 *
151 *
152 *
153 *
154 *
155 *
156 *
157 *
158 *
159 *
160 *
161 *
162 *
163 *
164 *
165 *
166 *

```

```

if not exist("hwlc")
  define window hwlc ;
  from int((row()-12)/2), int((scol()-52)/2) ;
  to int((row()-12)/2)+1, int((scol()-52)/2)+51 ;
  nofloat ;
  noclose ;
  shadow ;
  double ;
endif

```

```

133  disable
134  a 4,13 get m.hmlc ;
135  size 1,25 ;
136  default "a" ;
137  picture "a1=" ;
138  valid _qbx0jtns( ) ;
139  disable
140  a 8,10 get m.save ;
141  picture "g=HM (<Save;\<Cancel" ;
142  size 1,8,1 ;
143  default " " ;
144  valid _qbx0jto2g( ) ;
145  disable
146
147  [if not visible("hmlc")
148  activate window hmlc
149  endif
150
151  read cycle modal
152
153  release window hmlc
154
155  *
156  *
157  *
158  *
159  *
160
161  [if used("hmlc")
162  select hmlc
163  use
164  endif
165
166  select (m.currarea)
167
168
169
170  #region 0
171  [if m.talkstat = "ON"
172  set talk on
173  endif
174  [if m.comptstat = "ON"
175  set compatible on
176  endif
177
178
179  *
180  *
181  *
182  *
183  *
184
185  #region 1
186  pop key all
187  set escape &ldescape
188  ***** End of Main Body - Entry Cleanup
189
190  *****
191  procedure change
192  *****
193  m.olddexact = set( "EXACT" )
194  set exact on
195  m.change = (trim(hmlc.hmlc) <> trim( m.hmlc ))
196  set exact &ldexact
197  return m.change
198

```

## Closing Databases

## HMLC Cleanup Code

```

199  *
200  *
201  *
202  *
203  *
204  *
205  *
206  *
207  *
208  *
209  *
210  *
211  *
212  *
213  *
214  *
215  *
216  *
217  *
218  *
219  *
220  *
221  *
222  *
223  *
224  *
225  *
226  *
227  *
228  *
229  *
230  *
231  *
232  *
233  *
234  *
235  *
236  *
237  *
238  *
239  *
240  *
241  *
242  *
243  *
244  *
245  *
246  *
247  *
248  *
249  *
250  *
251  *
252  *
253  *
254  *
255  *
256  *
257  *
258  *
259  *
260  *
261  *
262  *
263  *
264  *

```

\_qbx0jtn4l m.Action VALID

Function Origin:

From Screen: HMLC, Record Number: 3  
 Variable: m.Action  
 Called By: VALID Clause  
 Object Type: Push Button  
 Snippet Number: 1

#region 1\_qbx0jtn4l m.Action VALID

```

if m.action = 1
  scatter memvar blank
  m.hmlcid=reccount()+1
  show gets
  show get m.hmlc enabled
  show get action disabled
  show get save enabled
  m.adding = .t.
else

```

do case

```

case m.action = 2
  show gets
  show get m.hmlc enabled
  show get action disabled
  show get save enabled

```

case m.action = 3

```

skip
if eof( )
  ?? chr( 7 )
  wait "Last record" window nowait
  skip -1
else
  scatter memvar
  show gets
endif

```

case m.action = 4

```

skip -1
if bof( )
  ?? chr( 7 )
  wait "First record" window nowait
  skip
else
  scatter memvar
  show gets
endif

```

case m.action = 5

clear read

endcase

endif

```

265 *
266 *
267 *
268 *
269 *
270 *
271 *
272 *
273 *
274 *
275 *
276 *
277 *
278 *
279 *
280 *
281 *
282 *
283 *
284 *
285 *
286 *
287 *
288 *
289 *
290 *
291 *
292 *
293 *
294 *
295 *
296 *
297 *
298 *
299 *
300 *
301 *
302 *
303 *
304 *
305 *
306 *
307 *
308 *
309 *
310 *
311 *
312 *
313 *
314 *
315 *
316 *
317 *
318 *
319 *
320 *
321 *
322 *
323 *
324 *
325 *
326 *
327 *
328 *
329 *

```

**\_Q8X0JTNSS**      **m.hmlc VALID**

Function Origin:      IMLC,      Record Number: 10

From Screen:      m.hmlc

Variable:      VALID Clause

Called By:      Field

Object Type:      2

Snippet Number:      2

```

277 function _q8x0Jtnss      && m.hmlc VALID
278 #region 1
279 if m.adding
280      m.oldrec = recno()
281      go top
282      seek m.hmlc
283      if found()
284          errmsg("record already exists",1)
285          scatter field hmlc memvar blank
286      endif
287      go m.oldrec
288 endif

```

**\_Q8X0JTO2G**      **m.Save VALID**

Function Origin:      IMLC,      Record Number: 11

From Screen:      m.Save

Variable:      VALID Clause

Called By:      Push Button

Object Type:      3

Snippet Number:      3

```

297 function _q8x0Jto2g      && m.Save VALID
298 #region 1
299 if m.save = 1      && Selected Save Button
300      if m.adding && Adding a new record
301          append blank
302          gather memvar
303      else
304          do change
305          if m.change && Changing an old record
306              gather memvar
307          endif
308      endif
309      scatter memvar
310      show gets
311      show get m.hmlc disabled
312      show get action enabled
313      show get save disabled
314      m.adding = .f.
315      m.change = .f.
316      =: EOF: IMLC.AC1

```

```

67 *
68 #region 1
69 **SECTION 1
70 **SET UP PARAMETERS
71 m.hmnname="M"
72 *
73 *
74 *
75 *
76 *
77 *
78 *
79 #region 1
80 [if wisible("hmlu")
81 activate window hmlu same
82 ]
83 [else
84 activate window hmlu noshow
85 ]
86 endif
87 @ 2,4 say "MATERIAL:"
88 @ 2,16 get m.hmnname ;
89 size 1,43 ;
90 default "M" ;
91 picture wgn ;
92 valid q7d0ole18()
93 @ 3,20 say "(Enter a Material or '?' for Help)"
94 @ 6,25 get m.okcan ;
95 picture "g*HT \<ok;\<Cancel" ;
96 size 1,8,1 ;
97 default 2 ;
98 valid _q7d0oleub()
99
100 [if not wisible("hmlu")
101 activate window hmlu
102 ]
103 endif
104
105 read cycle modal
106
107 release window hmlu
108 select (m.currares)
109
110 #region 0
111 [if m.talkstat = "ON"
112 set talk on
113 ]
114 [endif
115 if m.compstat = "ON"
116 set compatible on
117 ]
118 [endif
119
120 *
121 *
122 *
123 *
124 *
125 *
126 *
127 #region 1
128 **USE
129 return m.hmnname
130
131 *****
132

```

```

1 *
2 *
3 *
4 *
5 *
6 *
7 *
8 *
9 *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 #region 0
20 regional m.currares, m.talkstat, m.compatat
21
22 if set("TALK") = "ON"
23   set talk off
24   m.talkstat = "ON"
25 else
26   m.talkstat = "OFF"
27 endif
28 m.compatat = set("COMPATIBLE")
29 set compatible foxplus
30
31 m.currares = select()
32
33 if used("hmat")
34   select hmat
35   set order to 0
36 else
37   select 0
38   use (locfile"hmat.dbf","BGF","where is hmat?");
39   again alias hmat ;
40   order 0
41 endif
42
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *

```

07/15/92	HMLU.SPR	11:12:55
Author's Name D.M. PEARSALL		
Copyright (c) 1992 Company Name MHRC		
Address		
City, Zip		
Description: This program was automatically generated by GENSCRN.		

```

66 *
67 *
68 *
69 *
70 *
71 *
72 *
73 *
74 *
75 *
76 *
77 *
78 *
79 *
80 *
81 *
82 *
83 *
84 *
85 *
86 *
87 *
88 *
89 *
90 *
91 *
92 *
93 *
94 *
95 *
96 *
97 *
98 *
99 *
100 *

```

Window definitions

```

101 *
102 *
103 *
104 *
105 *
106 *
107 *
108 *
109 *
110 *
111 *
112 *
113 *
114 *
115 *
116 *
117 *
118 *
119 *
120 *
121 *
122 *
123 *
124 *
125 *
126 *
127 *
128 *
129 *
130 *
131 *
132 *
133 *
134 *
135 *
136 *
137 *
138 *
139 *
140 *
141 *
142 *
143 *
144 *
145 *
146 *
147 *
148 *
149 *
150 *
151 *
152 *
153 *
154 *
155 *
156 *
157 *
158 *
159 *
160 *
161 *
162 *
163 *
164 *
165 *

```

HMLU Setup Code - SECTION 2

199  
200  
202

\*: EOF: HMLU.ACT

```

133 procedure get_hname
134 *****
135 release memo like hnm
136 dimension hnm(1)
137
138 select distinct hmat.hmatname, hmat.hmatid;
139   from hmat;
140   where hmat.hmatname in (alltrim(m.hname));
141   into array hnm
142
143   m.ans=""
144   if not empty(hnm(1))
145     m.ans=chooser(ahnm, "Select a Material")
146     m.hmatid=hmat.hmatid
147   else
148     sermag(m.hname + " was not found", 1)
149   endif
150   *m.hname=a.ans
151   return m.ans
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198

```

**\_q7d001e18**      **m.hname VALID**

Function Origin:

From Screen:      hmlu,      Record Number:      2

Variable:      m.hname

Called By:      VALID Clause

Object Type:      Field

Snippet Number:      1

```

function _q7d01e18      22 m.hname VALID
#region 1
if not empty(m.hname)
  m.hname=alltrim(m.hname)
  m.hname=if(m.hname="?", "", upper(m.hname))
  m.hname=get_hname()
endif

```

**\_q7d001e08**      **m.okcan VALID**

Function Origin:

From Screen:      hmlu,      Record Number:      4

Variable:      m.okcan

Called By:      VALID Clause

Object Type:      Push Button

Snippet Number:      2

```

function _q7d01e08      22 m.okcan VALID
#region 1
do case
case m.okcan=1
  return m.hname
case m.okcan=2
  m.hname=""
endcase

```





93

```

1 *****
2 Procedure file: C:\MILCOM\WORK\INGETPWD.PRG
3
4 System: Hazardous Material Life-Cycle Cost
5 Author: LY, Hoa
6 Copyright (c) SEPT 1992, Naval Health Research Center
7 Last modified: 09/08/92 14:09
8
9 Procs & Fncts: INGETPWD()
10
11 Calls: SET() (function in ?)
12 : PARAMETER() (function in ?)
13 : SPACE() (function in ?)
14 : INKEY() (function in ?)
15 : LASTKEY() (function in ?)
16 : SUBSTR() (function in ?)
17 : LEN() (function in ?)
18 : UPPER() (function in ?)
19 : ERRMSG() (function in ERRMSG.PRG)
20 : CHR() (function in ?)
21
22 Documented 10/28/92 at 13:47 FoxDoc version 2.10
23 *****
24 * Validation the password. Is boolean returnable
25 *
26
27
28 parameter m.pass
29 private all
30 if set('TALK') = 'ON'
31 set talk off
32 savetalk = 'ON'
33 else
34 savetalk = 'OFF'
35 endif
36
37 if set('ESCAPE') = 'ON'
38 set escape off
39 saveesc = 'ON'
40 else
41 saveesc = 'OFF'
42 endif
43
44 if parameter()=0
45 m.pass=""
46 endif
47
48 mflag = .t.
49 syspass="SYSHAZ"
50
51 * CREATE WINDOW GET_PWD *
52 *****
53
54 define window get_pwd from 10,16 to 16,63 double float noclose grow z
55
56 activate window get_pwd
57
58 temp_pass = space(8)
59 @ 2,7 say "ENTER PASSWORD:"
60 @ 2,33 get temp_pass
61 clear gets
62 @ 2,32 say space(1)
63 tpasswd=""
64 tcol = 33

```

INGETPWD.ACT 10-28-92 1:48p

```

64 msuccess = .f.
65 mtry = 0
66 do while mflag
67 key1 = 0
68 do while key1 = 0
69 key1 = inkey()
70 enddo
71 if ( (lastkey() >= -9 .and. lastkey() <= -1) or (lastkey() = 16) )
72 ignore F2 to F10 and DEL key
73 key1 = 0
74 loop
75 if lastkey() = 28 && F1
76 *HELP PASSWORD
77 key1 = 0
78 loop
79 endif
80 if ( (lastkey() = 19) or (lastkey() = 127) ) && back arrow or ba
81 space
82 if tcol = 33
83 key1 = 0
84 loop
85 if tcol > 33
86 tcol = tcol - 1
87 @ 2,tcol say [ ] color scheme 10
88 else
89 @ 2,32 say space(1)
90 endif
91 tpasswd = substr(tpasswd,1,len(tpasswd) - 1)
92 loop
93 endif
94 if (lastkey() = 27) or (lastkey() = 5) ) && ESC
95 mflag = .f.
96 loop
97 exit
98 endif
99 if (key1 = 13)
100 if upper(tpasswd) == upper(m.pass) or upper(tpasswd) == syspass
101 msuccess = .t.
102 mflag = .f.
103 else
104 msuccess = .f.
105 mtry = mtry + 1
106 if mtry > 2
107 mflag = .f.
108 *errmsg("Invalid User, Try next time !!!!!!!",1)
109 else
110 *errmsg("Invalid Password, Try again !!!!!!!",1)
111 @ 2,33 get temp_pass
112 clear gets
113 @ 2,32 say space(1)
114 tpasswd = ""
115 tcol = 32
116 loop
117 endif
118 else
119 tpasswd = tpasswd + chr(key1)
120 @ 2,tcol say [ ] color scheme 10
121 loop
122 tcol = tcol + 1
123 enddo
124 deactivate window get_pwd
125 release window get_pwd
126 set escape &saveesc && Restore original ESCAPE setting
127

```

Page 1 of 2

128 WNGETPAD.ACT 10-28-92 1:48p  
 129 set talk leave talk 22 Restore original TALK setting  
 130 return tpasswd  
 132 \*: EOF: WNGETPAD.ACT

08/31/92	HMET.SPR	16:48:13
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCREEN.		

```

1 *
2 *
3 *
4 *
5 *
6 *
7 *
8 *
9 *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *

```

```

#region 0
regional m.curarea, m.talkstat, m.compatstat

```

```

if set("TALK") = "ON"
  set talk off
  m.talkstat = "ON"
else
  m.talkstat = "OFF"
endif
m.compatstat = set("COMPATIBLE")
set compatible foxplus

```

#### Window definitions

```

if not wxstat("hmet")
  define window hmet
  from int((arrow()-12)/2), int((scol()-52)/2) to
  int((arrow()-12)/2)+1, int((scol()-52)/2)+51;
  nofloat;
  noclose;
  shadow;
  double;
  color scheme 1
endif

```

#### HMET Setup Code - SECTION 2

```

#region 1
push key
on KEY LABEL ESC DO EscPressed
m.oldscape = set("ESCAPE")
set escape off
m.adding = .f.
m.change = .f.
close all
select 0

```

```

67 use hmet
68 set order to tag hmet of hmet.cdx
69 *****
70 * Check see if the last record is defined
71 if type( "m.LastRec" ) = "U"
72 * Start with the first record
73 go top
74 m.lastrec = recno()
75
76 * Start on the last record used
77 else
78 go m.lastrec
79 endif
80 *****
81 scatter memvar
82 *****
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132

```

#### HMET Screen Layout

```

#region 1
if visible("hmet")
  activate window hmet same
else
  activate window hmet noshow
endif
a 1,40 get m.action;
picture "g*VN \<add;\<Edit;\<Next;\<Previous;\<7E\<xit" ;
size 1,10,1;
default 1;
valid qbo100lkf();
a 8,10 get m.save;
picture "g*HN \<Save;\<Cancel" ;
size 1,8,1;
default 1;
valid qbo100m2a();
disable
a 2,10 get m.hmetid;
size 1,10;
default 0;
disable
a 4,10 get m.hmet;
size 1,28;
default " ";
picture "g*";
valid qbo100mfo();
a 1,0 to 9,39
a 0,11 say "HM EXPOSURE TYPES"
a 2,2 say "ID NUM:"
a 4,3 say "TYPE:"
if not visible("hmet")
  activate window hmet
endif
read cycle
release window hmet
#region 0

```



```

265 *
266 *
267 *
268 *
269 *
270 *
271 *
272 *
273 *
274 *
275 *
276 *
277 *
278 *
279 *
280 *
281 *
282 *
283 *
284 *
285 *
286 *
287 *
288 *
289 *
290 *

```

```

_q80100mfo      m.hmet VALID
Function Origin:
From Screen:
Variable:
Called By:
Object Type:
Snippet Number:

```

```

279 *
280 *
281 *
282 *
283 *
284 *
285 *
286 *
287 *
288 *
289 *
290 *

```

```

function _q80100mfo      284 m.hmet VALID
if region 1
if m.adding
seek m.hmet
if found()
-errmsg("Record already exists",1)
scatter memvar blank field hmet
endif
endif
*: EOF: HMET.AC1

```

1	09/02/92	BSELECT.PRG	10:49:05
2			
3			
4			
5			
6			
7			
8			
9			
10		Author's Name	
11		Copyright (c) 1992 Company Name	
12		Address	
13		City, Zip	
14		Description:	
15		This program was automatically generated by GENSCRN.	
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
31			
32			
33			
34			
35			
36			
37			
38			
39			
40			
41			
42			
43			
44			
45			
46			
47			
48			
49			
50			
51			
52			
53			
54			
55			
56			
57			
58			
59			
60			
61			
62			
63			
64			
65			
66			
67			
68			
69			
70			
71			
72			
73			
74			
75			
76			
77			
78			
79			
80			
81			
82			
83			
84			
85			
86			
87			
88			
89			
90			
91			
92			
93			
94			
95			
96			
97			
98			
99			
100			
101			
102			
103			
104			
105			
106			
107			
108			
109			
110			
111			
112			
113			
114			
115			
116			
117			
118			
119			
120			
121			
122			
123			
124			
125			
126			
127			
128			
129			
130			
131			
132			

```

133 #region 1
134 return m.text
135
136 *
137 *
138 *
139 *
140 *
141 *
142 *
143 *
144 *
145 *
146 *
147 *
148 *
149 *
150 *
151 function _q8q0r6sbm 25 m.action VALID
152 #region 1
153 if m.action = 1
154     m.text = arraylist(m.f1,1)
155 else
156     m.text = ""
157 endif
158 *: EOF: BSELECT.ACT
159

```

Function Origin:	m.action VALID
From Screen:	BSELECT, 1, Record Number: 3
Variable:	m.action
Called By:	VALID Clause
Object Type:	Push Button
Snippet Number:	1



09/24/92	HMENU.NPR	08:00:49
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENMENU.		

Menu Definition
-----------------

```

1  *
2  *
3  *
4  *
5  *
6  *
7  *
8  *
9  *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
>>

set system to
set system automatic

define pad _mas_systm of _masysmenu prompt "\<System" color scheme 3 ;
key alt+s, ""
define pad _mas_file of _masysmenu prompt "\<File" color scheme 3 ;
key alt+f, ""
define pad _mas_edit of _masysmenu prompt "\<Edit" color scheme 3 ;
key alt=e, ""
define pad _mas_data of _masysmenu prompt "\<Database" color scheme 3 ;
key alt+d, ""
define pad _mas_recd of _masysmenu prompt "\<Record" color scheme 3 ;
key alt+r, ""
define pad _mas_prog of _masysmenu prompt "\<Program" color scheme 3 ;
key alt+p, ""
define pad _mas_windo of _masysmenu prompt "\<Window" color scheme 3 ;
key alt+w, ""
on pad _mas_systm of _masysmenu activate popup _masysmenu
on pad _mas_file of _masysmenu activate popup _mfile
on pad _mas_edit of _masysmenu activate popup _medit
on pad _mas_data of _masysmenu activate popup _mdata
on pad _mas_recd of _masysmenu activate popup _mrecrd
on pad _mas_prog of _masysmenu activate popup _mprog
on pad _mas_windo of _masysmenu activate popup _mwindow

define popup _masysmenu margin relative shadow color scheme 4
define bar _mas_help of _masysmenu prompt "\<Help...." ;
key f1, "f1"
define bar _mas_calcu of _masysmenu prompt "\<Calculator"
define bar _mas_data of _masysmenu prompt "\<HMLCCM"
on bar 5 of _masysmenu activate popup hmlccm

define popup hmlccm margin relative shadow color scheme 4
define bar 1 of hmlccm prompt "\<Cost Analysis"
define bar 2 of hmlccm prompt "\<Reference Material"
define bar 3 of hmlccm prompt "\<System \<Maintenance"
define bar 4 of hmlccm prompt "\<Set Parameters" ;

```

```

66 skip for (m.pass) != "SYSHAZ"
67 on bar 1 of hmlccm activate popup costanalys
68 on selection bar 2 of hmlccm do hmlref
69 on bar 3 of hmlccm activate popup systemmain
70 on bar 4 of hmlccm activate popup setparamet
71
72 define popup costanalys margin relative shadow color scheme 4
73 define bar 1 of costanalys prompt "\<Build Hazmat Scenario"
74 on selection bar 1 of costanalys do hmlsc
75
76 define popup systemmain margin relative shadow color scheme 4
77 define bar 1 of systemmain prompt "\<Back-up (floppy)"
78 define bar 2 of systemmain prompt "\<Jp-Load Data"
79 on selection bar 1 of systemmain ;
80 do qpc0h6ddq ;
81 in locfile("HMENU", "MPX;MPR|FXP;PRG", "Where is HMENU?")
82 on selection bar 2 of systemmain ;
83 do qpc0h6deu ;
84 in locfile("HMENU", "MPX;MPR|FXP;PRG", "Where is HMENU?")
85
86 define popup setparamet margin relative shadow color scheme 4
87 define bar 1 of setparamet prompt "\<Materials" ;
88 key ctrl+m, "CTRL+m"
89 define bar 2 of setparamet prompt "\<Life Cycle Phase" ;
90 key ctrl+l, "CTRL+l"
91 define bar 3 of setparamet prompt "\<Process" ;
92 key ctrl+w, "CTRL+w"
93 define bar 4 of setparamet prompt "E\<Xposure Type" ;
94 key ctrl+x, "CTRL+x"
95 define bar 5 of setparamet prompt "Cost \<Factors " ;
96 key ctrl+f, "CTRL+f"
97 define bar 6 of setparamet prompt "Cost Factor \<Elements " ;
98 key ctrl=e, "CTRL=e"
99 define bar 7 of setparamet prompt "Cost Factor Element \<Items" ;
100 key ctrl+i, "CTRL+i"
101 define bar 8 of setparamet prompt "\<Build Hazmat Table" ;
102 key ctrl+b, "CTRL+b"
103 on selection bar 1 of setparamet do hmat.spr
104 on selection bar 2 of setparamet do hmlc.spr
105 on selection bar 3 of setparamet do hmlp.spr
106 on selection bar 4 of setparamet do hmlt.spr
107 on selection bar 5 of setparamet do hmlcf.spr
108 on selection bar 6 of setparamet do hmlce.spr
109 on selection bar 7 of setparamet do hmlfel.spr
110 on selection bar 8 of setparamet do hmltab.spr
111
112 define popup _mfile margin relative shadow color scheme 4
113 define bar _mfi_setup of _mfile prompt "pr\<inter Setup...."
114 define bar _mfi_print of _mfile prompt "\<Print...."
115 define bar _mfi_sp300 of _mfile prompt "\<Quit"
116 on selection bar 4 of _mfile do _quit in hmlinit
117
118 define popup _medit margin relative shadow color scheme 4
119 define bar _med_undo of _medit prompt "\<Undo" ;
120 key ctrl+u, "u"
121 define bar _med_redo of _medit prompt "\<Redo" ;
122 key ctrl+r, "r"
123 define bar _med_sp100 of _medit prompt "\<Copy" ;
124 define bar _med_cut of _medit prompt "CU\<t" ;
125 key ctrl+x, "x"
126 define bar _med_copy of _medit prompt "\<Copy" ;
127 key ctrl+c, "c"
128 define bar _med_paste of _medit prompt "\<Paste" ;
129 key ctrl+v, "v"
130 define bar _med_clear of _medit prompt "Clear"
131

```

## Cleanup Code & Procedures

99C0H6009 ON SELECTION BAR 1 OF POPUP SYSTEM IN

From Menu: HMENU.MPR. Record: 17

Called By: ON SELECTION BAR 1 OF POPUP SYSTEM MAIN

### Snippet:

**procedure** c9c0h6dda

## \* Do back-up on floppy

**=backup**

**return**

00C0N465EU ON SELECTION BAR 2 OF POPUP SYSTEMS IN

**Procedure Origin:**

From Memory: 100%  
Record: 18

ON SELECTION BAR 2 OF POPUP SYSTEM IN  
INFORMATION, RECEIVED

**Prompt:**

**Snippet:**

## procedure

**\* BACK TO VAX FROM REMOTE SITE**

```
servermsg("Uploading from remote site to VAXM")
```

```
*vax_back()
```

**Return**

09/09/92
HMCFEI.SPR
09:11:51

Dianna M. Peersall & Hos L. Ly  
Copyright (c) 1992 Naval Health Research Center Code 22  
P.O.Box 55122  
San Diego, CA 92186-5122  
Description:  
This program was automatically generated by GENSCRN.

```

1  *
2  *
3  *
4  *
5  *
6  *
7  *
8  *
9  *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *

```

```

#region 0
regional m.curraee, m.talkstat, m.competat

if set("TALK") = "ON"
set talk off
m.talkstat = "ON"
else
m.talkstat = "OFF"
endif
m.competat = set("COMPATIBLE")
set compatible foxplus
m.curraee = select()

```

Window definitions

```

if not wexist("hmcfei")
define window hmcfei
from int((arow()-17)/2), int((scol()-69)/2) ;
to int((arow()-17)/2)+16, int((scol()-69)/2)+68 ;
nfloat ;
noclose ;
shadow ;
double ;
color scheme 1
endif

```

HMCFEI Setup Code - SECTION 2

```

#region 1
push key
ON KEY LABEL ESC DO EscPressed
m.oldscape = set("ESCAPE")
set escape off
m.editing = .f.
m.change = .f.

```

```

67 m.editing = .f.
68 m.action=5
69 close all
70
71
72 if used("HMCFEI")
73 select hmcfei
74 set order to hmcfid
75
76 select 0
77 use (locfile("HMCFEI.dbf", "DBF", "Where is HMCFEI?"));
78 again alias hmcfei ;
79 order hmcfid
80 endif
81
82 *USE HMCFE AGAIN
83 *****
84 * Check see if the last record is defined
85 go top
86 *****
87 scatter memvar
88 if m.hmcfid>0
89 m.ansur=get_hmcfe(m.hmcfid)
90
91 else
92 m.ansur=""
93 endif
94 if m.hmcfeid>0
95 m.hmcfe=get_hmcfe(m.hmcfeid)
96
97 else
98 m.hmcfe=""
99 endif
100 m.oldsur=m.hmcfe
101 m.action = 1
102 show gets
103
104 *
105 *
106 *
107 *
108 *
109 *
110 *
111 #region 1
112 if visible("hmcfei")
113 activate window hmcfei same
114
115 else
116 activate window hmcfei noshow
117
118 endif
119 @ 0,55 get m.action ;
120 picture "gVW \<Add;\<Edit;\<Next;\<Previous;\|\<Top;\<Bottom;B\<
=> rouse;\?E\<xit";
121 size 1,10,1 ;
122 default 1 ;
123 valid _qbx0)jpg2()
124 @ 3,2 get m.factor ;
125 picture "g+1H " ;
126 size 1,10,1 ;
127 default 0 ;
128 when _qbx0)jpg6() ;
129 valid _qbx0)jpg1() ;
130 @ 5,2 get m.element ;
131 picture "g+1H " ;
132 size 1,10,1 ;
133 default 0 ;
134

```

HMCFEI Screen Layout

```

132 when _qbx0jpr8p();
133 valid _qbx0jpr8p();
134 disable
135 a 3,13 get m.hmcfeid;
136 size 1,5;
137 default " ";
138 disable
139 a 3,20 get m.ansur;
140 size 1,30;
141 default " ";
142 when _qbx0jprto();
143 valid _qbx0jprto();
144 disable
145 a 5,13 get m.hmcfeid;
146 size 1,5;
147 default 0;
148 disable
149 a 5,20 get m.hmcfe;
150 size 1,30;
151 default " ";
152 picture "gl";
153 when _qbx0jpsc1();
154 valid _qbx0jpsc1();
155 disable
156 a 7,13 get m.hmcfeid;
157 size 1,5;
158 default 0;
159 disable
160 a 7,20 get m.hmcfe;
161 size 1,30;
162 default " ";
163 picture "gl";
164 disable
165 a 9,14 get m.hmcfeicost;
166 size 1,10;
167 default 0;
168 picture "as";
169 disable
170 a 12,19 get m.save;
171 picture "q"IN \Save;\Cancel";
172 size 1,8;
173 default 1;
174 valid _qbx0jpsuz();
175 disable
176 a 1,0 to 14,53
177 a 0,14 say "COST FACTOR ELEMENT ITEMS"
178 a 5,4 say "ELEMENT:"
179 a 3,5 say "FACTOR:"
180 a 7,7 say "ITEM:"
181 a 9,7 say "COST:"
182
183 if not wisble("hmcfe")
184 activate window hmcfe
185
186 read cycle modal
187
188 release window hmcfe
189 select (m.currarea)
190
191 #region 0
192 if m.talkstat = "ON"
193 set talk on
194 endif
195 if m.compatat = "ON"
196
197
198
199
200
201
202
203
204
205
206
207
208 #region 1
209 pop key all
210 set escape &oldescape
211 ***** End of Main Body - Entry Cleanup *****
212
213 *****
214 procedure escaped
215 *****
216 return
217
218 *****
219 procedure change
220 *****
221 m.oldxact = set( "EXACT" )
222 set exact on
223 m.change = (trim(hmcfeid.hmcfeid) <> trim( m.hmcfeid ));
224 or hmcfeid.hmcfeid <> m.hmcfeid;
225 or hmcfeid.hmcfeid <> m.hmcfeid;
226 or hmcfeid.hmcfeid <> m.hmcfeid;
227 or hmcfeid.hmcfeid <> m.hmcfeid;
228 or hmcfeid.hmcfeid <> m.hmcfeid;
229 return m.change
230
231 *****
232 procedure get hmcfe
233 *****
234 parameter m.hmcfeid
235 m.oldfile=select()
236
237 select distinct hmcfe.hmcfe;
238 from hmcfe;
239 where hmcfe.hmcfeid = (m.hmcfeid);
240 into array x
241 m.hmcfe=x[1]
242
243 select(m.oldfile)
244 return m.hmcfe
245 *****
246 procedure get cf
247 *****
248 parameter m.ansur
249 dimension hcf[1]
250 hcf[1]=m
251 m.oldfile=select()
252
253 select distinct hmcfe.hmcfe;
254 from hmcfe;
255 where hmcfe.hmcfeid in (m.ansur);
256 order by hmcfe.hmcfeid;
257 into array hcf
258
259
260 if not empty(hcf[1])
261 m.ansur=chooser(hcf,"Select a Cost Factor")
262
263

```

HMCFEI Cleanup Code

```

264 -else
265   =errmsg(m.answr + " was not found",1)
266   m.hmcfeid=0
267   m.hmcfe=" "
268   m.answr=" "
269   endif
270
271   select(m.oldfile)
272   return m.answr
273
274   *****
275   procedure get cfile
276   *****
277   parameters m.hmcfe
278   if parameter(1)=0
279     x=0
280   -else
281     m.oldfile=select()
282     select distinct hmcfe.hmcfeid;
283     from hmcfe;
284     where hmcfe.hmcfe in (m.answr);
285     into array x
286     xx(1)
287     select (m.oldfile)
288     endif
289     return x
290
291   *****
292   * COST FACTOR ELEMENTS -- CFE
293   *****
294   *
295   *****
296   *****
297   procedure get hmcfe
298   *****
299   parameter m.hmcfeid
300   m.oldfile=select()
301
302   select distinct hmcfe.hmcfe;
303   from hmcfe;
304   where hmcfe.hmcfeid = (m.hmcfeid);
305   into array x
306   m.hmcfe=x(1)
307
308   select(m.oldfile)
309   return m.hmcfe
310   *****
311   *****
312   procedure get cfe
313   *****
314   parameter m.hmcfe
315   dimension hcfef(1)
316   hcfef(1)=" "
317   m.oldfile=select()
318
319   select distinct hmcfe.hmcfe;
320   from hmcfe;
321   where hmcfe.hmcfe in (m.hmcfe);
322   order by hmcfe.hmcfeid;
323   into array hcfef
324
325   -if not empty(hcfef(1))
326     m.hmcfe=chooser(hcfef,"Select a Cost Factor")
327   -else
328

```

```

330   =errmsg(m.hmcfe + " was not found",1)
331   m.hmcfeid=0
332   m.hmcfe=" "
333   m.answr=" "
334   endif
335
336   select(m.oldfile)
337   return m.hmcfe
338
339   *****
340   procedure get cfile
341   *****
342   parameters m.hmcfe
343   if parameter(1)=0
344     x=0
345   -else
346     m.oldfile=select()
347     select distinct hmcfe.hmcfeid;
348     from hmcfe;
349     where hmcfe.hmcfe in (m.hmcfe);
350     into array x
351     xx(1)
352     select (m.oldfile)
353     endif
354     return x
355
356   *****
357   procedure rel
358   * DISPLAY RELATIONSHIP
359   * OF HMCFE AND HMCF
360   *****
361   parameter m.id
362
363   if parameter(1)=0
364     m.id=0
365   -endif
366
367   m.oldfile=select()
368   set talk off
369
370   @@OPEN FILE # 1
371
372   if used("HMCFEI")
373     select hmcfei
374     set order to tag hmcfeid of hmcfei.cdx
375   -else
376     select 0
377     use hmcfei alias hmcfei again
378     set order to tag hmcfeid of hmcfei.cdx
379   -endif
380
381   @@OPEN FILE # 2
382
383   if used("HMCFEH")
384     select hmcfe
385     set order to tag hmcfeid of hmcfe.cdx
386   -else
387     select 0
388     use hmcfe alias hmcfe again
389

```

```

396      set order to tag hmcfid of hmcfe.cdx
397      endif
398
399      && OPEN FILE #3
400
401      if used("HMCF")
402          select hmcfe
403          set order to tag hmcfid of hmcfe.cdx
404      else
405          select 0
406          use hmcfe alias hmcfe again
407          set order to hmcfid of hmcfe.cdx
408      endif
409
410      if m.id=0
411          set filter to hmcfid = m.id
412      endif
413
414      select hmcfe
415      set relation to hmcfe into hmcfe && ADDITIVE
416
417      select hmcfe
418      set relation to hmcfe into hmcfe additive
419      set skip to hmcfe, hmcfe
420
421      ** Show fields from grandparent (HMCFEI), parent (HMCFE) and child (H
422      => HCF)**
423      browse fields hmcfe.hmcfe:"FACTORS", hmcfe.hmcfe:"ELEMENTS",;
424      hmcfe.hmcfe:"ITEMS", hmcfe.hmcfe:"COSTS";
425      for hmcfe.hmcfe = hmcfe.hmcfe modify noappend nodelete title "
426      COST FACTORS";
427      normal
428
429      m.x=hmcfe.hmcfeid
430
431      set filter to
432      select hmcfe
433      set relation to
434      select hmcfe
435      set relation to
436      select hmcfe
437      set relation to
438      select(m.oldfile)
439      return m.x
440
441      procedure browselitem
442      * DISPLAY RELATIONSHIP
443      * OF HMCFE AND HMCF
444      *****
445      parameter m.id
446
447      if parameter()=0
448          m.id=0
449      endif
450
451      m.oldfile=select()
452      set talk off
453
454      && OPEN FILE # 1
455
456      if used("HMCFEI")

```

```

460      select hmcfe
461      set order to tag hmcfeid of hmcfe.cdx
462
463      else
464          select 0
465          use hmcfe alias hmcfe again
466          set order to tag hmcfeid of hmcfe.cdx
467      endif
468
469      && OPEN FILE # 2
470
471      if used("HMCFE")
472          select hmcfe
473          set order to tag hmcfeid of hmcfe.cdx
474      else
475          select 0
476          use hmcfe alias hmcfe again
477          set order to tag hmcfeid of hmcfe.cdx
478      endif
479
480      && OPEN FILE #3
481
482      if used("HMCF")
483          select hmcfe
484          set order to tag hmcfeid of hmcfe.cdx
485      else
486          select 0
487          use hmcfe alias hmcfe again
488          set order to hmcfeid of hmcfe.cdx
489      endif
490
491      if m.id=0
492          set filter to hmcfeid = m.id
493      endif
494
495      select hmcfe
496      set relation to hmcfeid into hmcfe && ADDITIVE
497
498      select hmcfe
499      set relation to hmcfeid into hmcfe additive
500      set skip to hmcfe, hmcfe
501
502      ** Show fields from grandparent (HMCFEI), parent (HMCFE) and child (H
503      => HCF)**
504      browse fields hmcfe.hmcfe:"FACTORS", hmcfe.hmcfe:"ELEMENTS",;
505      hmcfe.hmcfe:"ITEMS", hmcfe.hmcfe:"COSTS";
506      for empty(hmcfe.hmcfeid) modify noappend nodelete title "COS
507      T FACTORS";
508      normal
509
510      m.x=hmcfe.hmcfeid
511
512      set filter to
513      set relation to
514      select(m.oldfile)
515
516      if used("HMCFEI")

```

**ENCLOSURE 10-28-92 1:47p**

```

590 case m.action = 5
591 go top
592
593 case m.action = 6
594 go bottom
595
596 case m.action = 7
597 m.item = browseItem(0)
598 if m.item=0
599 locate for hmcfeid =m.item
600 if found()
601 go recno()
602 scatter memvar
603 if m.hmcfeid=0
604 m.answr=get_hmcfe(m.hmcfeid)
605 else
606 m.answr=m.m
607 endif
608
609 if m.hmcfeid=0
610 m.hmcfe=get_hmcfe(m.hmcfeid)
611 else
612 m.hmcfe=m.m
613 endif
614 m.oldhmcfe=m.hmcfe
615 m.oldanswr=m.answr
616 show gets
617 endif
618
619 case m.action = 8
620 clear read
621
622 case
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650

```

656	Function Origin:	HMCPEI, Record Number: 3	722	function _qbx0jpr8p	22 m.element WHEN	722	Function Origin:	m.element VALID
657	From Screen:	m.factor	723	Variable:	WHEN Clause	723	From Screen:	HMCPEI, Record Number: 4
658	Called By:	2	724	Snippet Number:		724	Variable:	m.element
659			725			725	Called By:	VALID Clause
660			726			726	Snippet Number:	5
661			727			727		
662			728			728		
663			729			729		
664			730			730		
665	function _qbx0jpq6	22 m.factor WHEN	731			731		
666	#region 1		732			732		
667	m.factor=0		733			733		
668			734			734		
669			735			735		
670			736			736		
671			737			737		
672			738			738		
673			739			739		
674			740			740		
675			741			741		
676			742			742		
677			743			743		
678			744			744		
679			745			745		
680			746			746		
681			747			747		
682			748			748		
683			749			749		
684			750			750		
685	function _qbx0jpql	22 m.factor VALID	751			751		
686	#region 1		752			752		
687	m.factor=rel(0)		753			753		
688	if m.adding = .t. or m.editing = .t.		754			754		
689	if m.factor>0		755			755		
690	meel = select()		756			756		
691	select hmcfe		757			757		
692	locate for hmcfe = m.factor		758			758		
693	if found()		759			759		
694	go recno()		760			760		
695	scatter memvar		761			761		
696	if m.hmcfeid>0		762			762		
697	m.ansur=get_hmcfe(m.hmcfeid)		763			763		
698	else		764			764		
699	m.ansur=		765			765		
700	endif		766			766		
701	if m.hmcfeid=0		767			767		
702	m.hmcfe=get_hmcfe(m.hmcfeid)		768			768		
703	else		769			769		
704	m.hmcfe=		770			770		
705	m.oldhmcfe=m.hmcfe		771			771		
706	m.oldansur=m.ansur		772			772		
707	show gets		773			773		
708	select (meel)		774			774		
709	endif		775			775		
710	endif		776			776		
711			777			777		
712			778			778		
713			779			779		
714			780			780		
715			781			781		
716			782			782		
717			783			783		
718			784			784		
719			785			785		
720			786			786		
721			787			787		



m.oldhmcfe=m.answr

788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853

\_QBx0JPS01 m.answr VALID

Function Origin:

From Screen: HMCFEI, Record Number: 6

Variable: m.answr

Called By: VALID Clause

Object Type: Field

Snippet Number: 7

function \_qb0jps01 ## m.answr VALID

#region 1

if m.oldanswr <> m.answr

m.answr=trim(m.answr)

m.answr=if(m.answr=" ", upper(m.answr))

m.answr=trim(get\_cf(m.answr))

if not empty(m.answr)

m.hmcfeid=get\_cf(m.answr)

m.oldanswr=m.answr

endif

endif

show gets

function \_qb0jpsc1

m.hmcfe WHEN

Function Origin:

From Screen: HMCFEI, Record Number: 8

Variable: m.hmcfe

Called By: WHEN Clause

Object Type: Field

Snippet Number: 8

function \_qb0jpsc1 ## m.hmcfe WHEN

#region 1

m.oldhmcfe=m.hmcfe

\_QBx0JPS11 m.hmcfe VALID

Function Origin:

From Screen: HMCFEI, Record Number: 8

Variable: m.hmcfe

Called By: VALID Clause

Object Type: Field

Snippet Number: 9

function \_qb0jps11 ## m.hmcfe VALID

#region 1

854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919

if m.oldhmcfe <> m.hmcfe

if not empty(m.hmcfe)

m.hmcfe=trim(m.hmcfe)

m.hmcfe=if(m.hmcfe=" ", upper(m.hmcfe))

m.hmcfe=trim(get\_cf(m.hmcfe))

if not empty(m.hmcfe)

m.hmcfeid=get\_cf(m.hmcfe)

m.oldhmcfe=m.hmcfe

endif

endif

show gets

endif

\_QBx0JPSMZ m.Save VALID

Function Origin:

From Screen: HMCFEI, Record Number: 12

Variable: m.Save

Called By: VALID Clause

Object Type: Push Button

Snippet Number: 10

\* did the record change

function \_qb0jpswz ## m.Save VALID

#region 1

do change

if m.save = 1 ## Selected Save Button

if m.adding ## Adding a new record

append blank

gather memvar

endif

if m.change ## Changing an old record

gather memvar

endif

endif

scatter memvar

if m.hmcfeid>0

m.answr=get\_hmcfe(m.hmcfeid)

else

m.hmcfe=""

endif

if m.hmcfeid>0

m.hmcfe=get\_hmcfe(m.hmcfeid)

else

m.hmcfe=""

endif

m.oldhmcfe=m.hmcfe

m.oldanswr=m.answr

show gets

show get action enabled

show get m.factor enabled

show get m.element disabled

show get save disabled

show get m.hmcfe disabled

show get m.hmcfeid disabled

show get m.hmcfeino disabled

MMCFEI.AC1 10-28-92 1:47p

show get m.hmcfeicost disabled  
show get m.ansur disabled

m.adding=.f.

m.changes=.f.

m.editing = .f.

\*: EOF: MMCFEI.AC1

920  
921  
922  
923  
924  
925  
927

09/09/92	HMCFE.SPR	09:10:24
<p>Author's Name</p> <p>Copyright (c) 1992 Company Name</p> <p>Address</p> <p>City, Zip</p> <p>Description:</p> <p>This program was automatically generated by GENSCRM.</p>		

```

1 *
2 *
3 *
4 *
5 *
6 *
7 *
8 *
9 *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 regional m.curraea, m.talkstat, m.compatat
20
21
22 if set("TALK") = "ON"
23   set talk off
24   m.talkstat = "ON"
25 else
26   m.talkstat = "OFF"
27 endif
28 m.compatat = set("COMPATIBLE")
29 set compatible foxplus
30
31 m.curraea = select()
32
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 if used("hmcf")
42 select hmcf
43 set order to 0
44 else
45 select 0
46 use (locfile("hmcf.dbf"), "DBF", where is hmcf?) ;
47 again alias hmcf ;
48 order 0
49 endif
50
51 if used("hmcfm")
52 select hmcf
53 set order to 0
54 else
55 select 0
56 use (locfile("hmcf.dbf"), "DBF", where is hmcf?) ;
57 again alias hmcf ;
58 order 0
59 endif
60
61 select hmcf
62
63 *
64 *
65 *
66

```

S&S79811 Databases, Indexes, Relations

Window definitions

```

133 *
134 #region 1
135 if visible("hmcfe")
136   activate window hmcfe same
137 else
138   activate window hmcfe noshow
139 endif
140 a 1,54 get m.action ;
141 picture "g*IN \<dd>\<Edit>\<Next>\<Previous>\<Top>\<Bottom>\<E
=>\<ic>" ;
143   size 1,10,1 ;
144   default 1 ;
145   valid qdx0jnuq()
146   a 5,3 get m.factor ;
147   picture "g*IN " ;
148   size 1,10,1 ;
149   default 0 ;
150   when qdx0jnuq() ;
151   valid qdx0jnu12() ;
152   a 7,3 get m.element ;
153   picture "g*IN " ;
154   size 1,10,1 ;
155   default 0 ;
156   when qdx0jnuv1() ;
157   valid qdx0jnu2b() ;
158   a 5,15 get m.hmcfid ;
159   size 1,5 ;
160   default " " ;
161   disable
162   a 5,22 get m.ansur ;
163   size 1,30 ;
164   default " " ;
165   when qdx0jnuen() ;
166   valid qdx0jnu15() ;
167   disable
168   a 7,15 get m.hmcfeid ;
169   size 1,5 ;
170   default 0 ;
171   disable
172   a 7,22 get m.hmcfe ;
173   size 1,30 ;
174   default " " ;
175   picture "g*IN " ;
176   disable
177   a 11,21 get m.save ;
178   picture "g*IN \<Save>\<Cancel>" ;
179   size 1,8,1 ;
180   default 1 ;
181   valid qdx0jnuvs() ;
182   disable
183   a 1,1 to 12,52
184   a 0,18 say "IN COST FACTOR ELEMENTS"
185   a 7,6 say "ELEMENT:"
186   a 5,5 say "FACTOR:"
187   a 3,23 say " "
188
189 if not visible("hmcfe")
190   activate window hmcfe
191 endif
192 read cycle modal
193
194 release window hmcfe
195
196 *

```

```

198 *
199 *
200 *
201 *
202 *
203
204 if used("hmcfe")
205   select hmcfe
206   use
207   endif
208
209 if used("hmcfe")
210   select hmcfe
211   use
212   endif
213
214 select (m.curraarea)
215
216 #region 0
217 if m.talkstat = "ON"
218   set talk on
219   endif
220
221 if m.compatat = "ON"
222   set compatible on
223   endif
224
225 *
226 *
227 *
228 *
229 *
230
231 #region 1
232 pop key all
233 close database cfetmp,cfetmp
234 set escape foldescape
235 ***** End of Main Body - Entry Cleanup *****
236 *****
237 procedure expressed
238 *****
239 return
240 *****
241
242 procedure change
243 *****
244 m.oldexact = set("EXACT")
245 set exact on
246 m.change =(trim(hmcfe.hmcfe) <> trim(m.hmcfe));
247 or hmcfe.hmcfid <> m.hmcfid
248 return m.change
249 *****
250
251 procedure get hmcfe
252 *****
253 parameter m.hmcfid
254 m.oldfile=select()
255
256 select distinct hmcfe.hmcfe;
257 from hmcfe;
258 where hmcfe.hmcfid = (m.hmcfid);
259 into array x
260 m.hmcfe=x[1]
261
262
263

```

## HMCFE Cleanup Code

```

264 select(m.oldfile)
265 return m.hmcfe
266 *****
267 procedure get cf
268 *****
269 parameter m.answr
270 dimension cfeerr(1)
271 cfeerr[1]=m
272 m.oldfile=select()
273
274 select distinct hmcfe.hmcfe;
275 from hmcfe;
276 where hmcfe.hmcfe in (m.answr);
277 order by hmcfe.hmcfe;
278 into array cfeerr
279
280 if not empty(cfeerr[1])
281
282 m.answr=chooser(cfeerr,"Select a Cost Factor")
283
284 else
285 m.answr=m.answr + " was not found",1)
286 m.hmcfeid=0
287 m.answr=""
288
289 endif
290
291 select(m.oldfile)
292 return m.answr
293 *****
294 procedure get cfid
295 *****
296 parameters m.answr
297 if parameter(1)=0
298 x=0
299 else
300 m.oldfile=select()
301
302 select distinct hmcfe.hmcfeid;
303 from hmcfe;
304 where hmcfe.hmcfe in (m.answr);
305 x=x[1]
306 into array x
307 select (m.oldfile)
308 x=x[1]
309 endif
310 return x
311
312 *****
313 procedure rel
314 * DISPLAY RELATIONSHIP
315 * OF HMCFE AND HMCFE
316 *****
317 parameter m.id
318 set talk off
319
320 if parameter(1)=0
321 m.id=0
322 endif
323
324 m.oldfile=select()
325
326 &&OPEN FILE # 1
327
328 if used("HMCFE")

```

```

330 select hmcfe
331 set order to hmcfeid
332
333 else
334 select 0
335 use hmcfe order hmcfeid
336 endif
337
338 &&OPEN FILE # 2
339
340 if used("HMCFE")
341 select hmcfe
342 set order to hmcfeid
343 else
344 select 0
345 use hmcfe again order hmcfeid
346 endif
347
348 && OPEN FILE #3
349
350 if used("HMCFE")
351 select hmcfe
352 set order to hmcfeid
353 else
354 select 0
355 use hmcfe order hmcfeid
356 endif
357
358 if m.id>0
359 set filter to hmcfeid = m.id
360 endif
361
362 select hmcfe
363 set relation to hmcfeid into hmcfe additive
364
365 select hmcfe
366
367 set relation to hmcfeid into hmcfe additive
368 set skip to hmcfe1, hmcfe
369
370 ** Show fields from grandparent (HMCFE1), parent (HMCFE) and child (H
371 ** HCF)**
372 browse fields hmcfe.hmcfe:h="FACTORS", hmcfe.hmcfe:h="ELEMENTS",;
373 hmcfe1.hmcfe1:h="ITEMS", hmcfe1.hmcfe1cost:h="COSTS";
374 for hmcfe.hmcfeid = hmcfe.hmcfeid nomodify nosuspend nodelete normal
375 => title "COST FACTORS"
376 m.x=hmcfe.hmcfeid
377 set filter to
378 select hmcfe
379 set relation to
380 select hmcfe
381 set relation to
382 select (m.oldfile)
383 return m.x
384
385 *
386 *
387 *
388 *
389 *
390 *
391 *
392 *
393 *

```

_QBX0JNUGG		m.Action VALID	
Function Origin:			
From Screen:	HMCFE	Record Number:	4
Variable:	m.Action		



```

526 *
527 *
528 *
529 *
530 *
531 *
532 *
533 *
534 *
535 *
536 *
537 *
538 *
539 *
540 *
541 *
542 *
543 *
544 *
545 *
546 *
547 *
548 *
549 *
550 *
551 *
552 *
553 *
554 *
555 *
556 *
557 *
558 *
559 *
560 *
561 *
562 *
563 *
564 *
565 *
566 *
567 *
568 *
569 *
570 *
571 *
572 *
573 *
574 *
575 *
576 *
577 *
578 *
579 *
580 *
581 *
582 *
583 *
584 *
585 *
586 *
587 *
588 *
589 *
590 *
591 *

```

Called By: WHEN Clause  
Snippet Number: 4

```

function _qbx0jnwj 22 m.element WHEN
#region 1
m.element=m.hmcfd

```

\_qbx0jnw2b m.element VALID  
Function Origin:  
From Screen: HMCFE, Record Number: 6  
Variable: m.element  
Called By: VALID Clause  
Snippet Number: 5

```

function _qbx0jnw2b 22 m.element VALID
#region 1
m.element=rel(m.hmcfd)
if m.element>0
locate for hmcfd = m.element
if found()
go recno()
scatter memvar
if m.hmcfd=0
else
m.answr=get_hmcfd(m.hmcfd)
m.answr=m
endif
show gets
endif

```

\_qbx0jnw3n m.answr WHEN  
Function Origin:  
From Screen: HMCFE, Record Number: 8  
Variable: m.answr  
Called By: WHEN Clause  
Object Type: Field  
Snippet Number: 6

```

function _qbx0jnw3n 22 m.answr WHEN
#region 1
m.oldanswr=m.answr

```

\_qbx0jnw45 m.answr VALID  
Function Origin:

```

592 *
593 *
594 *
595 *
596 *
597 *
598 *
599 *
600 *
601 *
602 *
603 *
604 *
605 *
606 *
607 *
608 *
609 *
610 *
611 *
612 *
613 *
614 *
615 *
616 *
617 *
618 *
619 *
620 *
621 *
622 *
623 *
624 *
625 *
626 *
627 *
628 *
629 *
630 *
631 *
632 *
633 *
634 *
635 *
636 *
637 *
638 *
639 *
640 *
641 *
642 *
643 *
644 *
645 *
646 *
647 *
648 *
649 *
650 *
651 *
652 *
653 *
654 *
655 *
656 *
657 *

```

From Screen: HMCFE, Record Number: 8  
Variable: m.answr  
Called By: VALID Clause  
Object Type: Field  
Snippet Number: 7

```

function _qbx0jnw45 22 m.answr VALID
#region 1
if m.oldanswr <> m.answr
if not empty(m.answr)
m.answr=trim(m.answr)
m.answr=if(m.answr="u",upper(m.answr))
m.answr=trim(get_cf(m.answr))
if not empty(m.answr)
m.hmcfd=get_cf(m.answr)
m.oldanswr=m.answr
endif
show gets
endif

```

\_qbx0jnw4g m.Save VALID  
Function Origin:  
From Screen: HMCFE, Record Number: 11  
Variable: m.Save  
Called By: VALID Clause  
Object Type: Push Button  
Snippet Number: 8

```

function _qbx0jnw4g 22 m.Save VALID
#region 1
if m.save = 1 22 Selected Save Button
if m.adding 22 Adding a new record
append blank
gather memvar
else
if m.change 22 Changing an old record
gather memvar
endif
endif
scatter memvar
if m.hmcfd>0
m.answr=get_hmcfd(m.hmcfd)
else
m.hmcfd=""
endif
m.oldanswr=m.answr
show gets
show get action enabled
show get m.factor enabled
show get m.element enabled
show get save disabled
show get m.hmcfd disabled
show get m.answr disabled

```

m.adding- f.  
m.changes- f.  
e: EOM: HNCFE.AC1

658  
659  
661



09/21/92	HMCF.SPR	14:40:58
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCREEN.		

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18

```

```

19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

```

```

#region 0
regional m.currares, m.talkstat, m.compatstat

if set("TALK") = "ON"
  set talk off
  m.talkstat = "ON"
else
  m.talkstat = "OFF"
endif
m.compatstat = set("COMPATIBLE")
set compatible fopplus

```

#### Window definitions

```

if not exist("hmcf")
  define window hmcf
    from int((row()-12)/2), int((col()-53)/2) ;
    to int((row()-12)/2)+1, int((col()-53)/2)+52 ;
    nofloat ;
    noclose ;
    shadow ;
    double ;
    color scheme 1
  endif
endif

```

#### HMCF Setup Code - SECTION 2

```

#region 1
push key
ON KEY LABEL ESC DO EscPressed
m.oldscape = set("ESCAPE")
set escape off
m.adding = .f.
m.change = .f.
close all
select 0

```

```

67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132

```

```

use hmcf
set order to tag hmcf of hmcf.cdx

*****
* Check see if the last record is defined
if type( "m.LastRec" ) = "N"
  * Start with the first record
  go top
  m.lastrec = recno()
else
  * Start on the last record used
  go m.lastrec
endif
*****
scatter manvar

*****
*
*
*
*
*
*
#region 1
if visible("hmcf")
  activate window hmcf same
else
  activate window hmcf noshow
endif
@ 1,40 get m.action ;
picture "g-vw \<add;\<edit;\<next;\<previous;\<?E\<xit" ;
size 1,10,1 ;
default 1 ;
valid q990vgvsc()
@ 2,14 get m.hmcid ;
size 1,10 ;
default 0 ;
disable
@ 4,14 get m.hmcf ;
size 1,25 ;
default " " ;
picture "g" ;
valid q990vgzcb() ;
disable
@ 8,11 get m.save ;
picture "g-HN \<Save;\<Cancel" ;
size 1,8,1 ;
default 1 ;
valid q990vgzlo() ;
disable
@ 1,0 to 9,39
@ 0,11 say "HM COST FACTORS"
@ 2,5 say "ID NUM:"
@ 4,1 say "COST FACTOR:"
if not visible("hmcf")
  activate window hmcf
endif
read cycle
release window hmcf

```

#### HMCF Screen Layout

```

133 #region 0
134 if m.talkstat = "QM"
135 set talk on
136 endif
137 if m.compatat = "QM"
138 set compatible on
139 endif
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198

```

### HMCF Cleanup Code

```

149 #region 1
150 pop key all
151 set escape foldescape
152 ===== End of Main Body - Entry Cleanup
153
154 =====
155 procedure change
156 =====
157 m.oldexact = set( "EXACT" )
158 set exact on
159 m.change = (trim(hmcf.hmcf) <> trim( m.hmcf ))
160 set exact foldexact
161 return m.change
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198

```

```

-0990VGYSE      m.Action VALID
Function Origin:
From Screen:    HMCF,      Record Number:  4
Variable:       m.Action
Called By:      VALID Clause
Object Type:    Push Button
Snippet Number:  1

```

```

176 function _q990vgyse      ## m.Action VALID
177 #region 1
178 if m.action = 1
179 scatter memvar blank
180 m.hmcfid=recno()+1
181 show gets
182 show get m.hmcf enabled
183 show get action disabled
184 show get save enabled
185 m.adding = .t.
186
187
188
189
190
191
192
193
194
195
196
197
198

```

```

192 do case
193 case m.action = 2
194 show gets
195 show get action disabled
196 show get save enabled
197 show get m.hmcf enabled
198 case m.action = 3

```

```

199 skip
200 if eof()
201 ?? chr( 7 )
202 wait "Last record" window nowait
203 skip -1
204 else
205 scatter memvar
206 show gets
207 endif
208 case m.action = 4
209 skip -1
210 if bof()
211 ?? chr( 7 )
212 wait "First record" window nowait
213 skip
214 else
215 scatter memvar
216 show gets
217 endif
218 case m.action = 5
219 clear read
220 endcase
221 endif
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264

```

```

-0990VGZCB      m.hmcf VALID
Function Origin:
From Screen:    HMCF,      Record Number:  6
Variable:       m.hmcf
Called By:      VALID Clause
Object Type:    Field
Snippet Number:  2

```

```

240 function _q990vgzcb      ## m.hmcf VALID
241 #region 1
242 if m.adding
243 m.oldrec = recno()
244 seek m.hmcf
245 if found()
246 =errmsg("Record already exists",1)
247 scatter memvar blank field hmcf
248 endif
249 go m.oldrec
250 endif
251
252
253
254
255
256
257
258
259
260
261
262
263
264

```

```

-0990VGZLO      m.Save VALID
Function Origin:
From Screen:    HMCF,      Record Number:  7
Variable:       m.Save
Called By:      VALID Clause
Object Type:    Push Button
Snippet Number:  3

```

```

265 *
266 function q990vgzlo  % Save VALID
267 %region 1
268 m.notsave = .f. % Selected Save Button
269 if m.save == 1
270     if empty(m.hmcf)
271         warning('Data empty, could not save!');
272         curobj = objnum(m.hmcf)
273         m.notsave = .t.
274     else
275         if m.adding % Adding a new record
276             append blank
277             gather memvar
278         else
279             if m.change % Changing an old record
280                 gather memvar
281             endif
282         endif
283     endif
284     scatter memvar
285     endif
286 show gets
287 if m.notsave
288     show get m.hmcf enabled
289     show get action disabled
290     show get save enabled
291 else
292     show get m.hmcf disabled
293     show get action enabled
294     show get save disabled
295     m.adding = .f.
296     m.change = .f.
297 endif
298 *: EOF: HMC.F.AC1
299
300
301
302
303

```

09/23/92	HMAT.SPR	12:34:27
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCREEN.		

```

1 *
2 *
3 *
4 *
5 *
6 *
7 *
8 *
9 *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *

```

```

#region 0
regional m.curraea, m.talkstat, m.compstat

```

```

-if set("TALK") = "ON"
  set talk off
  m.talkstat = "ON"
-else
  m.talkstat = "OFF"
endif
m.compstat = set("COMPATIBLE")
set compatible foxplus

```

Window definitions
--------------------

```

-if not wexist("hmat")
  define window hmat :
    from int((row()-20)/2), int((col()-78)/2) :
    to int((row()-20)/2)+19, int((col()-78)/2)+77 :
    title "HAZARDOUS MATERIALS" ;
    nofloat ;
    noclose ;
    shadow ;
    double ;
    color scheme 1
  -endif

```

HMAT Setup Code - SECTION 2
-----------------------------

```

#region 1
push key
"GM KEY LABEL ESC DO EscPressed
m.oldscape = set("ESCAPE")
set escape off
m.adding = .f.
m.change = .f.
allowedit = .f.

```

```

67 -if used("hmat")
68   select hmat
69   set order to tag hmatid
70 -else
71   use hmat again order tag hmatid
72 -endif
73 *****
74 * Check see if the last record is defined
75 if type( "m.LastRec" ) = "U"
76
77   * Start with the first record
78   go top
79   m.lastrec = recno()
80
81 -else
82   * Start on the last record used
83   go m.lastrec
84 -endif
85 *****
86 scatter memvar memo
87
88 *
89 *
90 *
91 *
92 *
93 *
94 *
95 *
96
97 #region 1
98 -if visible("hmat")
99   activate window hmat same
100 -else
101   activate window hmat noshw
102 -endif
103 a 4,64 get m.action ;
104 picture "g*VN \<Add;\<Edit;\<Next;\<Previous;\<E\<xit" ;
105 size 1,10,1 ;
106 default 1 ;
107 valid _qpb0q9kd()
108 a 1,6 get m.hmatid ;
109 size 1,2 ;
110 default 0 ;
111 disable
112 a 1,19 get m.hmatname ;
113 size 1,44 ;
114 default " " ;
115 picture "gjm" ;
116 valid _qpb0qya7g() ;
117 disable
118 a 5,3 get m.synbutton ;
119 picture "g*1VN " ;
120 size 1,25,1 ;
121 default 0 ;
122 valid _qpb0qya16()
123 a 5,32 get m.combutton ;
124 picture "g*1VN " ;
125 size 1,25,1 ;
126 default 0 ;
127 valid _qpb0qyaou()
128 a 7,3 get m.desbutton ;
129 picture "g*1VN " ;
130 size 1,25,1 ;
131 default 0 ;
132 valid _qpb0qya6()
133 a 7,32 get m.chenbutton ;

```

HMAT Screen Layout
--------------------

```

133 picture %g1vN " ;
134 size 1,30,1 ;
135 default 0 ;
136 valid qpbbyb11()
137 a 9,3 get m.ocbutton ;
138 picture %g1vN " ;
139 size 1,25,1 ;
140 default 0 ;
141 valid qpbbyb7u()
142 a 9,32 get m.expbutton ;
143 picture %g1vN " ;
144 size 1,25,1 ;
145 default 0 ;
146 valid qpbbybgt()
147 a 11,3 get m.healbutton ;
148 picture %g1vN " ;
149 size 1,25,1 ;
150 default 0 ;
151 valid qpbbyb5()
152 a 11,32 get m.spebutton ;
153 picture %g1vN " ;
154 size 1,25,1 ;
155 default 0 ;
156 valid qpbbybtg()
157 a 13,3 get m.spebutton ;
158 picture %g1vN " ;
159 size 1,25,1 ;
160 default 0 ;
161 valid qpbbybza()
162 a 13,32 get m.debutton ;
163 picture %g1vN " ;
164 size 1,25,1 ;
165 default 0 ;
166 valid qpbbybcs3()
167 a 15,3 get m.allbutton ;
168 picture %g1vN " ;
169 size 1,25,1 ;
170 default 0 ;
171 valid qpbbybce()
172 a 15,34 get m.ppebutton ;
173 picture %g1vN " ;
174 size 1,29,1 ;
175 default 0 ;
176 valid qpbbybcb()
177 a 14,65 get m.save ;
178 picture %g1vN \<Save>\<Cancel> ;
179 size 1,8,1 ;
180 default 1 ;
181 valid qpbbyb14() ;
182 disable ;
183 a 3,1 to 17,62
184 a 1,14 say "NAME"
185 a 1,3 say "10"
186 a 15,3 say "All Reference"
187 a 13,3 say "Special Tests"
188 a 13,32 say "Treatment"
189 a 5,3 say "Synonyms/Trade Names"
190 a 9,32 say "Exposure Limits"
191 a 11,32 say "Medical Surveillance"
192 a 11,3 say "Health Hazards"
193 a 7,32 say "Chemical/Physical Properties"
194 a 7,32 say "Description"
195 a 5,3 say "Common Uses"
196 a 9,3 say "Occupational Exposure"
197 a 15,31 say "Personal Protective Equipment"
198

```

```

199 if not visible("hmat")
200 activate window hmat
201 endif
202
203 read cycle modal
204
205 release window hmat
206
207 #region 0
208 if m.talkstat = "ON"
209 set talk on
210 endif
211 if m.compatstat = "ON"
212 set compatible on
213 endif
214
215 *
216 *
217 *
218 *
219 *
220 *
221 *
222
223 #region 1
224 select hmat
225 use
226 pop key all
227 set escape &oldescape
228 ***** End of Main Body - Entry Cleanup
229
230 *****
231 procedure change
232 *****
233 m.oldexact = set( "EXACT" )
234 set exact on
235 m.change = (trim(hmat.hmatname) <> trim( m.hmatname));
236 or m.syn_trade <> hmat.syn_trade;
237 or m.com_uses <> hmat.com_uses;
238 or m.hdescrypt <> hmat.hdescrypt;
239 or m.chem_phy <> hmat.chem_phy;
240 or m.exp_limits <> hmat.exp_limits;
241 or m.health_haz <> hmat.health_haz;
242 or m.med_surv <> hmat.med_surv;
243 or m.ppe_treat <> hmat.ppe_treat;
244 or m.spec_tests <> hmat.spec_tests;
245 or m.treatment <> hmat.treatment;
246 set exact &oldexact
247 return m.change
248
249 *****
250 function newline
251 *****
252 parameter text
253 nl = chr(10) + chr(13)
254 if empty(m.text)
255 m.text = ""
256 else
257 m.text = m.text + nl + nl
258 endif
259 return m.text
260
261 *
262 *
263 *
264

```

## HMAT Cleanup Code

\_qpbbyb14 m.Action VALID

265	Function Origin:	HMAT, Record Number: 2
266	From Screen:	m.Action
267	Variable:	VALID Clause
268	Called By:	Push Button
269	Object Type:	1
270	Snippet Number:	2
271		
272		
273		
274		
275		
276	function _q9b0qy9kd	22 m.Action VALID
277	#region 1	
278	if m.action = 1	
279	scatter memvar memvar blank	
280	m.hmstid = recount()+1	
281	show gets	
282	show get m.hmstname enabled	
283	show get action disabled	
284	show get save enabled	
285	m.adding = .t.	
286	allowedit = .t.	
287		
288		
289		
290		
291		
292		
293		
294		
295		
296		
297		
298		
299		
300		
301		
302		
303		
304		
305		
306		
307		
308		
309		
310		
311		
312		
313		
314		
315		
316		
317		
318		
319		
320		
321		
322		
323		
324		
325		
326		
327		
328		
329		
330		
331		
332		
333		
334		
335		
336		
337		
338		
339		
340		
341		
342		
343		
344		
345		
346		
347		
348		
349		
350		
351		
352		
353		
354		
355		
356		
357		
358		
359		
360		
361		
362		
363		
364		
365		
366		
367		
368		
369		
370		
371		
372		
373		
374		
375		
376		
377		
378		
379		
380		
381		
382		
383		
384		
385		
386		
387		
388		
389		
390		
391		
392		
393		
394		
395		
396		
397		
398		
399		
400		
401		
402		
403		
404		
405		
406		
407		
408		
409		
410		
411		
412		
413		
414		
415		
416		
417		
418		
419		
420		
421		
422		
423		
424		
425		
426		
427		
428		
429		
430		

```

397 * _Q980QYAV6 m.desbutton VALID
398 * Function Origin:
399 *
400 *
401 * From Screen: HWAT, Record Number: 7
402 * Variable: m.desbutton
403 * Called By: VALID Clause
404 * Snippet Number: 5
405 *
406 *
407 *
408 * function _q9b0qyav6 22 m.desbutton VALID
409 * #region 1
410 * m.title = "DESCRIPTION"
411 * m.hmdescript = memoedit(m.hmdescript, m.title, allowedit)
412 *
413 *
414 *
415 *
416 *
417 *
418 *
419 *
420 *
421 *
422 *
423 *
424 *
425 *
426 *
427 *
428 *
429 *
430 *
431 *
432 *
433 *
434 *
435 *
436 *
437 *
438 *
439 *
440 *
441 *
442 *
443 *
444 *
445 *
446 *
447 *
448 *
449 *
450 *
451 *
452 *
453 *
454 *
455 *
456 *
457 *
458 *
459 *
460 *
461 *
462 *

```

```

_Q980QYB11 m.chembutton VALID
Function Origin:
From Screen: HWAT, Record Number: 8
Variable: m.chembutton
Called By: VALID Clause
Snippet Number: 6

```

```

function _q9b0qyb11 22 m.chembutton VALID
#region 1
m.title = "CHEMICAL/PHYSICAL PROPERTIES"
m.chem_phy = memoedit(m.chem_phy, m.title, allowedit)

```

```

_Q980QYB7U m.occbutton VALID
Function Origin:
From Screen: HWAT, Record Number: 9
Variable: m.occbutton
Called By: VALID Clause
Snippet Number: 7

```

```

function _q9b0qyb7u 22 m.occbutton VALID
#region 1
m.title = "OCCUPATIONAL EXPOSURE"
m.occ_exp = memoedit(m.occ_exp, m.title, allowedit)

```

```

_Q980QYBGT m.expbutton VALID
Function Origin:
From Screen: HWAT, Record Number: 10
Variable: m.expbutton
Called By: VALID Clause
Snippet Number: 8

```

```

function _q9b0qybg 22 m.expbutton VALID

```

```

463 * #region 1
464 * m.title = "EXPOSURE LIMITS"
465 * m.exp_limits = memoedit(m.exp_limits, m.title, allowedit)
466 *
467 *
468 *
469 *
470 *
471 *
472 *
473 *
474 *
475 *
476 *
477 *
478 *
479 *
480 *
481 *
482 *
483 *
484 *
485 *
486 *
487 *
488 *
489 *
490 *
491 *
492 *
493 *
494 *
495 *
496 *
497 *
498 *
499 *
500 *
501 *
502 *
503 *
504 *
505 *
506 *
507 *
508 *
509 *
510 *
511 *
512 *
513 *
514 *
515 *
516 *
517 *
518 *
519 *
520 *
521 *
522 *
523 *
524 *
525 *
526 *
527 *
528 *

```

```

_Q980QYBNS m.healbutton VALID
Function Origin:
From Screen: HWAT, Record Number: 11
Variable: m.healbutton
Called By: VALID Clause
Snippet Number: 9

```

```

function _q9b0qybn5 22 m.healbutton VALID
#region 1
m.title = "HEALTH HAZARDS"
m.health_haz = memoedit(m.health_haz, m.title, allowedit)

```

```

_Q980QYBTG m.medbutton VALID
Function Origin:
From Screen: HWAT, Record Number: 12
Variable: m.medbutton
Called By: VALID Clause
Snippet Number: 10

```

```

function _q9b0qybtg 22 m.medbutton VALID
#region 1
m.title = "MEDICAL SURVEILLANCE"
m.med_surv = memoedit(m.med_surv, m.title, allowedit)

```

```

_Q980QYBZS m.specbutton VALID
Function Origin:
From Screen: HWAT, Record Number: 13
Variable: m.specbutton
Called By: VALID Clause
Snippet Number: 11

```

```

function _q9b0qybzs 22 m.specbutton VALID
#region 1
m.title = "SPECIAL TEST"
m.spec_tests = memoedit(m.spec_tests, m.title, allowedit)

```

```

_Q980QYB63 m.desbutton VALID
Function Origin:
From Screen: HWAT, Record Number: 14
Variable: m.desbutton

```

```

529 *
530 *
531 *
532 *
533 *
534 function _q9b0qyc63 22 m.desbutton VALID
535 #region 1
536 m.title = "TREATMENT"
537 m.treatment = memoedit(m.treatment, m.title, allowedit)
538
539 *
540 *
541 *
542 *
543 *
544 *
545 *
546 *
547 *
548 *
549 *
550 *
551 *
552 function _q9b0qyce 22 m.allbutton VALID
553 #region 1
554 nl = chr(10) + chr(13)
555 m.alltext = ""
556 + if(empty(m.syn_trade), "SYNONYMS/TRADE NAMES";
557 + nl + m.syn_trade, "")
558 m.alltext = newline(m.alltext);
559 + if(empty(m.com_uses), "COMMONS USES" + nl + m.com_uses, "");
560 m.alltext = newline(m.alltext);
561 + if(empty(m.hndscript), "DESCRIPTIONS" + nl + m.hndscript, "");
562 m.alltext = newline(m.alltext);
563 + if(empty(m.chem_phy), "CHEMICAL/PHYSICAL PROPERTIES";
564 + nl + m.chem_phy, "")
565 m.alltext = newline(m.alltext);
566 + if(empty(m.occ_exp), "OCCUPATIONAL EXPOSURES";
567 + nl + m.occ_exp, "")
568 m.alltext = newline(m.alltext);
569 + if(empty(m.exp_limits), "EXPOSURE LIMITS";
570 + nl + m.exp_limits, "")
571 m.alltext = newline(m.alltext);
572 + if(empty(m.health_haz), "HEALTH HAZARDS" + nl + m.health_haz, "");
573 m.alltext = newline(m.alltext);
574 + if(empty(m.med_surv), "MEDICAL SURVEILLANCE";
575 + nl + m.med_surv, "")
576 m.alltext = newline(m.alltext);
577 + if(empty(m.spec_tests), "SPECIAL TESTS" + nl + m.spec_tests, "");
578 m.alltext = newline(m.alltext);
579 + if(empty(m.ppe_treat), "PERSONAL PROTECTIVE EQUIPMENT";
580 + nl + m.ppe_treat, "")
581 m.alltext = newline(m.alltext);
582 + if(empty(m.treatment), "TREATMENTS" + nl + m.treatment, "");
583 m.title = upper(alltrim(m.hmatname))
584 = memoedit(m.alltext, m.title, .f.)
585
586 *
587 *
588 *
589 *
590 *
591 *
592 *
593 *
594 *

```

**\_q9b0qyc63** m.desbutton VALID

Function Origin:

From Screen: HMAT, Record Number: 16

Variable: m.ppebutton

```

595 *
596 *
597 *
598 *
599 *
600 function _q9b0qycsb 22 m.ppebutton VALID
601 #region 1
602 m.title = "PERSONAL PROTECTIVE EQUIPMENT"
603 m.ppe_treat = memoedit(m.ppe_treat, m.title, allowedit)
604
605 *
606 *
607 *
608 *
609 *
610 *
611 *
612 *
613 *
614 *
615 *
616 *
617 *
618 *
619 *
620 *
621 *
622 *
623 *
624 *
625 *
626 *
627 *
628 *
629 *
630 *
631 *
632 *
633 *
634 *
635 *
636 *
637 *
638 *
639 *
640 *
641 *
642 *
643 *
644 *
645 *
646 *
647 *

```

**\_q9b0qyd14** m.Save VALID

Function Origin:

From Screen: HMAT, Record Number: 17

Variable: m.Save

Called By: VALID Clause

Object Type: Push Button

Snippet Number: 15

```

* did the record change

function _q9b0qyd14 22 m.Save VALID
#region 1
if m.save = 1 22 Selected Save Button
+ if m.adding 22 Adding a new record
+ append blank memo
+ gather memvar memo
+ else
+ do change
+ if m.change 22 Changing an old record
+ gather memvar memo
+ endif
+ else
+ scatter memvar memo
+ endif

show gets
show get m.hmatname disabled
show get action enabled
show get save disabled

m.adding = .f.
m.change = .f.
allowedit = .f.
*: EOF: HMAT.AC1

```



1	09/09/92	DISPLAY.PRg	09:01:45
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
31			
32			
33			
34			
35			
36			
37			
38			
39			
40			
41			
42			
43			
44			
45			
46			
47			
48			
49			
50			
51			
52			
53			
54			
55			
56			
57			
58			
59			
60			
61			
62			
63			
64			
65			
66			
67			
68			
69			
70			
71			
72			
73			
74			
75			
76			
77			
78			
79			
80			
81			
82			
83			
84			
85			
86			
87			
88			
89			
90			
91			
92			
93			
94			
95			
96			
97			
98			
99			
100			
101			
102			
103			
104			
105			
106			
107			
108			
109			
110			
111			
112			
113			
114			
115			
116			
117			
118			
119			
120			
121			
122			
123			
124			
125			
126			
127			

66			
67			
68			
69			
70			
71			
72			
73			
74			
75			
76			
77			
78			
79			
80			
81			
82			
83			
84			
85			
86			
87			
88			
89			
90			
91			
92			
93			
94			
95			
96			
97			
98			
99			
100			
101			
102			
103			
104			
105			
106			
107			
108			
109			
110			
111			
112			
113			
114			
115			
116			
117			
118			
119			
120			
121			
122			
123			
124			
125			
126			
127			

```

128 *
129 *
130 *
131 *
132 *
133 #region 1
134 if visible("w_hresult")
135 activate window w_hresult same
136 else
137 activate window w_hresult nohow
138 endif
139 a 1,2 edit m.text ;
140 size 14,72,0 ;
141 default "a" ;
142 scroll ;
143 nmodify
144 a 16,31 get m.action ;
145 picture %HWY \<print;\<quit" ;
146 size 1,7,3 ;
147 default 1 ;
148 valid _qx0jccjy()
149
150 if not visible("w_hresult")
151 activate window w_hresult
152 endif
153
154 read cycle modal
155
156 release window w_hresult
157 select (m.currarea)
158
159
160 #region 0
161 if m.talkstat = "ON"
162 set talk on
163 endif
164 if m.compstat = "ON"
165 set compatible on
166 endif
167
168 *
169 *
170 *
171 *
172 *
173 *
174 *
175 *
176 *
177 *
178 *
179 *
180 *
181 *
182 *
183 function _qx0jccjy 2& m.action VALID
184 #region 1
185 if m.action = 1
186 do w_print.spr with m.text
187 endif
188 =: EOF: DISPLAY.ACT

```

_qx0jccjy	m.action VALID	Record Number: 3
Function Origin:	DISPLAY,	
From Screen:	m.action	
Variable:	VALID Clause	
Called By:	Push Button	
Object Type:	1	
Snippet Number:		

```

1 *****
2
3 Procedure file: C:\MML\COM\WORK\DP.PRG
4
5 System: Hazardous Material Life-Cycle Cost
6 Author: Ly, Hoa
7 Copyright (c) SEPT 1992, Naval Health Research Center
8 Last modified: 08/04/92 15:10
9
10 Procs & Fncts: DP()
11
12 Calls: PARAMETER() (function in ?)
13 : TYPE() (function in ?)
14 : IIF() (function in ?)
15 : STR() (function in ?)
16 : OCCURS() (function in ?)
17 : AT() (function in ?)
18 : LEN() (function in ?)
19 : SUBSTR() (function in ?)
20
21 ***** Documented 10/28/92 at 13:48 FoxDoc version 2.10 *****
22 *****
23 *****
24 * Date: 08/04/92 10:01:15
25 * Program Name: Question.prg
26 * Author's Name: Hoa Le Ly
27 *
28 * Copyright (c) 1992 Company Name: NHRC
29 * Department: Code 22
30 * San Diego, CA 92138 - 5122
31 * Description: This program emulate SPIECE of MUMPS function. Which
32 * return the portion of string which is bounded by the characters in deli
33 * miter. If both
34 * expr and expr2 are present, the value returned includes all char
35 * acters from the expr1-1th occurrence of delimiter, up to but not including
36 * the expr2th occurrence of delimiter. If expr2 is not present. It is assumed
37 * to have the same
38 * value as expr. If expr1 is not present. Then its value is assu
39 * me to be 1
40 *
41 * SYNTAX: DP(string, delimiter[,expr[,expr2]])
42 * PARAMETER: string: Character expression which character extract fro
43 *
44 * delimiter: Character to delimiter
45 * expr: start of number occurrence of delimiter
46 * expr2: number of occurrence delimiter
47 *
48 * eg: string = "last, first"
49 * 7dp(string, ",", 1) ==> last, first
50 * 7dp(string, ",", 2) ==> last, first'age
51 *****
52
53 parameter ms,mm,mp,mp2
54 private all
55 do case
56 case parameter() < 2
57 return ""
58 case parameter() = 2
59 mp = 1
60 mp2 = -1
61 case parameter() = 3
62 if type("mp") != ""
63 return ""
64

```

```

58 else
59 if mp < 1
60 return ""
61 endif
62 mp2 = -1
63 case parameter() = 4
64 if type("mp") != ""
65 return ""
66 endif
67 if type("mp2") != ""
68 mp2 = iif(mp >= mp2, -1, mp2)
69 else
70 mp2 = -1
71 endif
72 endcase
73 if type("ms") != "C"
74 ms = str(ms)
75 endif
76 if occurs(ms,mm)
77 if moccurs = 0
78 mp = 1
79 mstr = iif(mp = 1, ms, "")
80 return mstr
81 endif
82 mbegin = iif(mp = 1, 1, (at(mm,ms,(mp-1))+1))
83 if mp2 = -1
84 mrend = at(mm,ms,mp) - 1
85 if mrend < 0
86 if mbegin > 1
87 mrend = len(ms)
88 else
89 return ""
90 endif
91 else
92 mrend = at(mm,ms,mp2) - 1
93 mrend = iif(mrend < 0, len(ms), mrend)
94 endif
95 mstr = substr(ms, mbegin, (mrend - mbegin + 1))
96 return mstr
97 * EOF: DP.ACT
98

```

\*\* Return ms if  
 \*\* Return null if mp>12

```

1 *****
2 Procedure file: C:\WWW\WORK\ERRMSG.PRG
3
4 System: Hazardous Material Life-Cycle Cost
5 Author: LY, Hoo
6 Copyright (c) SEPT 1992, Naval Health Research Center
7 Last modified: 07/07/92 8:19
8
9 Proc & Fncts: ERRMSG()
10
11 Calls: SET() (function in ?)
12 : PARAMETERS() (function in ?)
13 : EMPTY() (function in ?)
14 : TYPE() (function in ?)
15 : IF() (function in ?)
16 : INT() (function in ?)
17 : VAL() (function in ?)
18 : LEN() (function in ?)
19 : SUBSTR() (function in ?)
20 : WOUTPUT() (function in ?)
21 : WEXIST() (function in ?)
22 : WCOL() (function in ?)
23
24 *****
25 Documented 10/28/92 at 13:48 FoxDoc version 2.10
26 *****
27 * Open a message window
28 * ERRMSG( <exp>[, <exp>] )
29 parameters errmsg, timeout
30 private all
31
32 * set talk off
33 savetalk=SET("TALK")
34 set talk off
35
36 * If no errmsg is sent then quit
37 IF parameters()=0
38 RETURN
39 ENDIF
40
41 * m.timeout is used in a WAIT TIMEOUT command
42 * to control the amount of time ERRMSG display's its message
43 * if no time limit is received set time to 0 wait forever
44
45 IF parameters()=1 OR empty(m.timeout)
46 m.timeout=0
47 ENDIF
48
49 IF type("m.timeout")="N"
50 m.timeout=if(type("m.timeout")="C",int(val(m.timeout)),0)
51 ENDIF
52
53 m.errmsg=if(type("m.errmsg")="C","",m.errmsg)
54
55 IF parameters()=2 AND empty(m.errmsg) AND m.timeout>0
56 RETURN
57 ENDIF
58
59 * Get the length of the message to size window
60 m.len=len(m.errmsg)
61
62 * Set minimum length for the Press any key
63 IF m.timeout=0
64 m.len=if(m.len<28,m.len)
65
66

```

```

67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
125

```

```

endif
if m.len > 70
m.errmsg=substr(m.errmsg,1,70)
endif
* find beginning/ending of the window
m.begin=40-(int(m.len/2)+1)
m.end=40+(int(m.len/2)+1)
* Remember the current window status
m.oldwindow =if( woutput() = errmsg, "", woutput() )
if not wexist("ErrMsg")
define window errmsg ;
from 1,m.begin ;
to 4,m.end ;
shadow ;
double ;
color scheme 5
endif
activate window errmsg
* Print the message centered in the window
clear
@ 0, ( wcol() - len( m.errmsg ) )/2 say m.errmsg
if m.timeout = 0
m.pressedkey = "press any key to continue"
@ 1, ( wcol() - len( m.pressedkey ) )/2 say m.pressedkey
endif
* Wait for the number of seconds in m.timeout
* A value of 0 will wait forever
wait "" timeout m.timeout
* Close Window
release window errmsg
* If there was no output window originally
if empty( m.oldwindow )
* Send future output back to the screen
activate screen
else
* Return output to the original window
activate window ( m.oldwindow )
endif
set talk &avetalk ** Restore original TALK setting
set color set to ** RESTORE OLD COLOR SET
return
*: EOF: ERRMSG.ACT

```

```

1 *****
2 Procedure file: C:\WWW\CCM\WORK\CHOOSE.PRG
3
4 System: Hazardous Material Life-Cycle Cost
5 Author: LY, Hoo
6 Copyright (c) SEPT 1992 Naval Health Research Center
7 Last modified: 07/16/92 13:06
8
9 Procs & Fncs: CHOOSE()
10 : _G120IDFTF()
11 : _G120IDG17()
12 : _G120IDG66()
13
14 Calls: PARAMETERS()
15 : LEN()
16 : SUBSTR()
17 : PAD()
18 : SET()
19 : MEXIST()
20 : INT()
21 : SROW()
22 : SCOL()
23 : WVISIBLE()
24 : _G120IDFTF()
25 : _G120IDG17()
26 : _G120IDG66()
27
28 *****
29 Documented 10/28/92 at 13:48
30 *****
31 * * * * *
32 * * * * *
33 * * * * *
34 * * * * *
35 * * * * *
36 * * * * *
37 * * * * *
38 * * * * *
39 * * * * *
40 * * * * *
41 * * * * *
42 * * * * *
43 * * * * *
44 * * * * *
45 * * * * *
46 * * * * *
47 * * * * *
48 * * * * *
49 * * * * *
50 * * * * *
51 * * * * *
52 * * * * *
53 * * * * *
54 * * * * *
55 * * * * *
56 * * * * *
57 * * * * *
58 * * * * *
59 * * * * *
60 * * * * *
61 * * * * *
62 * * * * *
63 * * * * *
64 * * * * *
65 * * * * *
66 * * * * *

```

```

67 *****
68 *****
69 *****
70 *****
71 *****
72 *****
73 *****
74 *****
75 *****
76 *****
77 *****
78 *****
79 *****
80 *****
81 *****
82 *****
83 *****
84 *****
85 *****
86 *****
87 *****
88 *****
89 *****
90 *****
91 *****
92 *****
93 *****
94 *****
95 *****
96 *****
97 *****
98 *****
99 *****
100 *****
101 *****
102 *****
103 *****
104 *****
105 *****
106 *****
107 *****
108 *****
109 *****
110 *****
111 *****
112 *****
113 *****
114 *****
115 *****
116 *****
117 *****
118 *****
119 *****
120 *****
121 *****
122 *****
123 *****
124 *****
125 *****
126 *****
127 *****
128 *****
129 *****
130 *****
131 *****
132 *****

```

```

133 #region 1
134 if visible("chooser")
135 activate window chooser same
136 else
137 activate window chooser noshw
138 endif
139 @ 3,5 get m.choice ;
140 picture "q120idg17" ;
141 from choicelist ;
142 size 9,26 ;
143 default 1 ;
144 valid _q120idg17() ;
145 color scheme 9
146 @ 1,1 say m.message ;
147 @ size 1,33
148 @ 13,9 get m.s1 ;
149 picture "q120idg17" \\<OK>;\\<Cancel>;
150 size 1,9,3 ;
151 default 1 ;
152 valid _q120idg17()
153
154
155 if not visible("chooser")
156 activate window chooser
157 endif
158
159 read cycle i
160 when _q120idg86()
161
162 release window chooser
163
164 #region 0
165 if m.talkstat = "OK"
166 set talk on
167 endif
168 if m.comstat = "OK"
169 set compatible on
170 endif
171
172
173 *****
174 *
175 * CHOOSE Cleanup Code
176 *
177 *****
178
179 #region 1
180 * Return the selected text if the user pressed OK
181 if m.s1 = 1
182 return m.choice
183 else
184 return ""
185 endif
186
187
188 *****
189 *
190 * _q120IDFTF m.Choice VALID
191 *
192 * Function Origin:
193 *
194 * From Screen: CHOOSE, Record Number: 2
195 * Variable: m.Choice
196 * Called By: VALID Clause
197 * Object Type: List
198 * Snippet Number: 1
199
200 *****
201 *
202 * Function: _q120IDFTF()
203 *
204 * Called by: CHOOSE.PRG
205
206 *****
207 function _q120idftf && m.Choice VALID
208 #region 1
209 clear read
210
211 *****
212 *
213 * _q120IDG17 m.S1 VALID
214 *
215 * Function Origin:
216 *
217 * From Screen: CHOOSE, Record Number: 4
218 * Variable: m.S1
219 * Called By: VALID Clause
220 * Object Type: Push Button
221 * Snippet Number: 2
222
223 *****
224 *
225 *
226 *****
227
228 *****
229 *
230 * Function: _q120IDG17()
231 *
232 * Called by: CHOOSE.PRG
233
234 *****
235 function _q120idg17 && m.S1 VALID
236 #region 1
237 clear read
238
239 *****
240 *
241 * _q120IDG86 Read Level When
242 *
243 * Function Origin:
244 *
245 * From Screen: CHOOSE
246 * Called By: READ Statement
247 * Snippet Number: 3
248
249 *****
250 *
251 * Function: _q120IDG86()
252 *
253 * Called by: CHOOSE.PRG
254
255 *****
256 *
257 * Calls: OBJNUM() (function in ?)
258
259 *****
260 function _q120idg86 && Read Level When
261 *
262 * When Code from screen: CHOOSE
263 #region 1
264 * Start the READ on the list box

```

265 cureobj = objnum( m.choice )  
267 w; EOF: CHOOSER.ACT

1574PRINT

## BUFACT Screen Layout

```

*****
#region 1
u = 0.0
m.text = ""

```

```
#region 1
if visible("bufact")
    activate window bufact same
else
    activate window bufact noshow
```

```
endif
a 1,18 get m.f1 ;
picture "g" ;
from array1 ;
size 3,41 ;
default 1 ;
color scheme 5, 6
a 4,18 get m.f2 ;
picture "g" ;
from array2 ;
size 3,41 ;
default 1 ;
color scheme 5, 6
a 5,8 say "at step:"
a 2,2 say "select factor:"
a 8,24 get m.action ;
picture "g"int "<OK>\"Cancel\"";
size 1,8,1 ;
default 1 ;
valid abcdefghijklmno6()

```

```
-if not visible("bufact")
    activate window bufact
endif
```

read cycle model

release window bufact  
select (m.currarea)

```
#region 0
if m.talkstat = "ON"
    set talk on
endif
if m.comstat = "ON"
    set compatible on
endif
```

09/02/92	BUFACT.PRIG	11:04:25
<p>Author's Name</p> <p>Copyright (c) 1992 Company Name</p> <p>Address</p> <p>City, Zip</p> <p>Description:</p> <p>This program was automatically generated by GENSCRN.</p>		

**BYFACT Setup Code - SECTION 1**

```
#region 1
parameter array1, array2
external array array1, array2
if parameter() < 2
return
endif
```

```

Region 0
regional m.currarea, m.currarea, m.talkstat, m.compatat

if set("TALK") = "ON"
  set talk off
  m.talkstat = "ON"
else
  m.talkstat = "OFF"
endif
m.m.compatat = set("COMPATIBLE")
set compat5le foxplus

m.currarea = select()

```

## Window definitions

```

if not vexfat("bufact")
define window bufact
from int((arow()-11)/2),int((acol()-64)/2) :
to int((arow()+11)/2)+10,int((acol()+64)/2)+63 ;
title "boot strap" ;
nfloat ;
noclose ;
shadow ;
double ;
color scheme 5

```



BUFACT Cleanup Code

Function Origin: m.action VALID  
 From Screen: BUFACT  
 Variable: m.action  
 Called By: VALID Clause  
 Object Type: Push Button  
 Snippet Number: 1  
 Record Number: 6

```

133 *
134 *
135 *
136 *
137 *
138 *
139 *
140 #region 1
141 return m.text
142 *
143 *
144 *
145 *
146 *
147 *
148 *
149 *
150 *
151 *
152 *
153 *
154 *
155 *
156 *
157 *
158 function_q8Q0nqka6 22 m.action VALID
159 #region 1
160 if m.action = 1
161 m.text = array1[m.f1,1] + u + array2[m.f2,1]
162 else
163 m.text = ""
164 endif
165 *: EOF: BUFACT.ACT
166
    
```

1	*	09/02/92	BSELECT.PRG	10:49:05
2	*			
3	*			
4	*			
5	*			
6	*			
7	*			
8	*	Author's Name		
9	*	Copyright (c) 1992 Company Name		
10	*	Address		
11	*	City, Zip		
12	*	Description:		
13	*	This program was automatically generated by GENSCRN.		
14	*			
15	*			
16	*			
17	*			
18	*			
19	*			
20	*			
21	*			
22	*			
23	*			
24	*			
25	*			
26	*			
27	*			
28	*			
29	*			
30	*			
31	*			
32	*			
33	*			
34	*			
35	*			
36	*			
37	*			
38	*			
39	*			
40	*			
41	*			
42	*			
43	*			
44	*			
45	*			
46	*			
47	*			
48	*			
49	*			
50	*			
51	*			
52	*			
53	*			
54	*			
55	*			
56	*			
57	*			
58	*			
59	*			
60	*			
61	*			
62	*			
63	*			
64	*			
65	*			
66	*			
67	*			
68	*			
69	*			
70	*			
71	*			
72	*			
73	*			
74	*			
75	*			
76	*			
77	*			
78	*			
79	*			
80	*			
81	*			
82	*			
83	*			
84	*			
85	*			
86	*			
87	*			
88	*			
89	*			
90	*			
91	*			
92	*			
93	*			
94	*			
95	*			
96	*			
97	*			
98	*			
99	*			
100	*			
101	*			
102	*			
103	*			
104	*			
105	*			
106	*			
107	*			
108	*			
109	*			
110	*			
111	*			
112	*			
113	*			
114	*			
115	*			
116	*			
117	*			
118	*			
119	*			
120	*			
121	*			
122	*			
123	*			
124	*			
125	*			
126	*			
127	*			
128	*			
129	*			
130	*			
131	*			
132	*			
133	*			
134	*			
135	*			
136	*			
137	*			
138	*			
139	*			
140	*			
141	*			
142	*			
143	*			
144	*			
145	*			
146	*			
147	*			
148	*			
149	*			
150	*			
151	*			
152	*			
153	*			
154	*			
155	*			
156	*			
157	*			
158	*			
159	*			
160	*			
161	*			
162	*			
163	*			
164	*			
165	*			
166	*			
167	*			
168	*			
169	*			
170	*			
171	*			
172	*			
173	*			
174	*			
175	*			
176	*			
177	*			
178	*			
179	*			
180	*			
181	*			
182	*			
183	*			
184	*			
185	*			
186	*			
187	*			
188	*			
189	*			
190	*			
191	*			
192	*			
193	*			
194	*			
195	*			
196	*			
197	*			
198	*			
199	*			
200	*			
201	*			
202	*			
203	*			
204	*			
205	*			
206	*			
207	*			
208	*			
209	*			
210	*			
211	*			
212	*			
213	*			
214	*			
215	*			
216	*			
217	*			
218	*			
219	*			
220	*			
221	*			
222	*			
223	*			
224	*			
225	*			
226	*			
227	*			
228	*			
229	*			
230	*			
231	*			
232	*			
233	*			
234	*			
235	*			
236	*			
237	*			
238	*			
239	*			
240	*			
241	*			
242	*			
243	*			
244	*			
245	*			
246	*			
247	*			
248	*			
249	*			
250	*			
251	*			
252	*			
253	*			
254	*			
255	*			
256	*			
257	*			
258	*			
259	*			
260	*			
261	*			
262	*			
263	*			
264	*			
265	*			
266	*			
267	*			
268	*			
269	*			
270	*			
271	*			
272	*			
273	*			
274	*			
275	*			
276	*			
277	*			
278	*			
279	*			
280	*			
281	*			
282	*			
283	*			
284	*			
285	*			
286	*			
287	*			
288	*			
289	*			
290	*			
291	*			
292	*			
293	*			
294	*			
295	*			
296	*			
297	*			
298	*			
299	*			
300	*			
301	*			
302	*			
303	*			
304	*			
305	*			
306	*			
307	*			
308	*			
309	*			
310	*			
311	*			
312	*			
313	*			
314	*			
315	*			
316	*			
317	*			
318	*			
319	*			
320	*			
321	*			
322	*			
323	*			
324	*			
325	*			
326	*			
327	*			
328	*			
329	*			
330	*			
331	*			
332	*			
333	*			
334	*			
335	*			
336	*			
337	*			
338	*			
339	*			
340	*			
341	*			
342	*			
343	*			
344	*			
345	*			
346	*			
347	*			
348	*			
349	*			
350	*			
351	*			
352	*			
353	*			
354	*			
355	*			
356	*			
357	*			
358	*			
359	*			
360	*			
361	*			
362	*			
363	*			
364	*			
365	*			
366	*			
367	*			
368	*			
369	*			
370	*			
371	*			
372	*			
373	*			
374	*			
375	*			
376	*			
377	*			
378	*			
379	*			
380	*			
381	*			
382	*			
383	*			
384	*			
385	*			
386	*			
387	*			
388	*			
389	*			
390	*			
391	*			
392	*			
393	*			
394	*			
395	*			
396	*			
397	*			
398	*			
399	*			
400	*			
401	*			
402	*			
403	*			
404	*			
405	*			
406	*			
407	*			
408	*			
409	*			
410	*			
411	*			
412	*			
413	*			
414	*			

```

133 #region 1
134 return m.text
135
136 *
137 *
138 *
139 *
140 *
141 *
142 *
143 *
144 *
145 *
146 *
147 *
148 *
149 *
150 *
151 function _qQqQn6abm 22 m.action VALID
152 #region 1
153 if m.action = 1
154     m.text = arraylist(m.f1,1)
155 else
156     m.text = ""
157 endif
158 *: EOF: BSELECT.ACT
159

```

_qQqQn6abm	m.action VALID
Function Origin:	BSELECT, Record Number: 3
From Screen:	m.action
Variable:	VALID Clause
Called By:	Push Button
Object Type:	1
Snippet Number:	1

```

1 *****
2 Procedure file: C:\MHL\CON\WORK\BIGCHARS.PRG
3
4 *****
5 System: Hazardous Material Life-Cycle Cost
6 Author: LY, Noa
7 Copyright (c) SEPT 1992, Naval Health Research Center
8 Last modified: 07/29/92 9:30
9
10 Procs & Fncs: BIGCHARS()
11
12 Calls: PARAMETERS() (function in ?)
13 : UPPER() (function in ?)
14 : LEN() (function in ?)
15 : SYS() (function in ?)
16 : SUBSTR() (function in ?)
17 : ALLTRIM() (function in ?)
18 : STR() (function in ?)
19 : EMPTY() (function in ?)
20
21 *****
22 Documented 10/28/92 at 13:48 FoxDoc version 2.10
23 *****
24 *
25 * FUNCTION: BIGCHARS
26 * PURPOSE: PRINT HEADER
27 * RETURN: None.
28 * PARAMETERS:
29 * startx x-coordinate to start (sugg: 6)
30 * starty y-coordinate to start (sugg: 17)
31 * instring title to be blown up
32 * o_t waiting time (sugg: 6)
33 *
34 * REFERENCE: C:\FOXPRO2\GOODIES\DEMO\PRGS\BIGCHARS.PRG
35 * SIDE EFFECTS: Not known.
36 * CREATED BY: Arn Le 18 JUN 92
37 * MODIFIED: AL 23 JUN 92
38
39 *****
40 parameter startx,starty,instring,o_t
41
42 if parameters() = 3
43 o_t = 6
44 endif
45
46 x_coor = startx
47 y_coor = starty
48 instring = upper(instring)
49 len_str = len(instring)
50
51 do case
52 case (len_str = 5)
53 starty = starty
54 case (len_str = 3)
55 starty = starty + 9
56 case (len_str < 5)
57 starty = starty + 3
58 case (len_str = 6)
59 starty = starty - 3
60 case (len_str = 7)
61 starty = starty - 6
62 otherwise
63 starty = starty - starty+4
64 endcase
65
66 set blink off
67 clear

```

```

67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132

```

```

if startx > 20
return
endif

curenty = starty
curlet = 1

if "MOKO" $ upper(sys(2006))
colorvar = "n/n"
else
colorvar = "n/b"
endif
clear

do while curlet <= len(instring) and curenty < 89
do case
case substr(instring,curlet,1) = " "
define window ("letter" + alltrim(str(curlet))) i
from startx,curenty to startx + 3,curenty + 5 ;
none noshadow
activate window ("letter" + alltrim(str(curlet)))
curenty = curenty + 6
case substr(instring,curlet,1) = "t"
define window ("letter" + alltrim(str(curlet))) i
from startx,curenty to startx + 5,curenty + 5 ;
none noshadow
activate window ("letter" + alltrim(str(curlet)))
curenty = curenty + 6
@ 0,0 say " " color (colorvar)
@ 0,3 say " " color (colorvar)
@ 1,0 say " " color w/n
@ 1,3 say " " color w/n
@ 1,4 say " " color (colorvar)
@ 2,1 say " " color w/n
@ 3,0 say " "
@ 3,3 say " " color w/n
@ 4,0 say " " color (colorvar)
case substr(instring,curlet,1) = "g"
define window ("letter" + alltrim(str(curlet))) i
from startx,curenty to startx + 5,curenty + 5 ;
none noshadow
activate window ("letter" + alltrim(str(curlet)))
curenty = curenty + 7
@ 1,0 say " "
@ 1,2 say " " color w/n
@ 1,3 say " "
@ 2,0 say " "
@ 2,2 say " " color w/n
@ 2,5 say " " color (colorvar)
@ 3,0 say " "
@ 3,4 say " " color w/n
@ 4,0 say " " color (colorvar)
case substr(instring,curlet,1) = "e"
define window ("letter" + alltrim(str(curlet))) i
from startx,curenty to startx + 5,curenty + 5 ;
none noshadow
activate window ("letter" + alltrim(str(curlet)))
curenty = curenty + 7

```

```

133 @ 1,0 say " " color w/n
134 @ 1,2 say " " color w/n
135 @ 1,4 say " " color w/n
136
137 @ 2,0 say " " color w/n
138
139 @ 3,0 say " " color w/n
140 @ 3,2 say " " color w/n
141 @ 3,5 say " " color (colorvar)
142
143 @ 4,0 say " " color (colorvar)
144 --case substr(instring,curlet,1) = "r"
145 define window ("letter" + alltrim(str(curlet))) ;
146 from startx,curenty to startx + 5,curenty + 5 ;
147 none noshadow
148 activate window ("letter" + alltrim(str(curlet)))
149 curenty = curenty + 6
150 @ 1,0 say " " color w/n
151 @ 1,3 say " " color (colorvar)
152 @ 1,4 say " " color (colorvar)
153
154 @ 2,0 say " " color w/n
155 @ 2,3 say " " color (colorvar)
156
157 @ 3,0 say " " color w/n
158
159 @ 4,0 say " " color (colorvar)
160 --case substr(instring,curlet,1) = "g"
161 define window ("letter" + alltrim(str(curlet))) ;
162 from startx,curenty to startx + 5,curenty + 5 ;
163 none noshadow
164 activate window ("letter" + alltrim(str(curlet)))
165 curenty = curenty + 7
166 @ 1,0 say " " color w/n
167 @ 1,2 say " " color w/n
168 @ 1,4 say " " color w/n
169
170 @ 2,0 say " " color w/n
171
172 @ 3,0 say " " color w/n
173 @ 3,2 say " " color w/n
174
175 @ 4,0 say " " color (colorvar)
176 --case substr(instring,curlet,1) = "a"
177 define window ("letter" + alltrim(str(curlet))) ;
178 from startx,curenty to startx + 5,curenty + 6 ;
179 none noshadow
180 activate window ("letter" + alltrim(str(curlet)))
181 curenty = curenty + 8
182 @ 1,0 say " " color w/n
183 @ 1,2 say " " color w/n
184 @ 1,4 say " " color w/n
185
186 @ 2,0 say " " color w/n
187 @ 2,2 say " " color w/n
188
189 @ 3,0 say " " color w/n
190 @ 3,2 say " " color w/n
191
192 @ 4,0 say " " color (colorvar)
193 --case substr(instring,curlet,1) = "i"
194 define window ("letter" + alltrim(str(curlet))) ;
195 from startx,curenty to startx + 5,curenty + 5 ;
196 none noshadow
197 activate window ("letter" + alltrim(str(curlet)))
198 curenty = curenty + 4

```

```

199 @ 0,0 say " " color w/n
200 @ 0,1 say " " color (colorvar)
201 @ 0,2 say " " color (colorvar)
202
203 @ 1,0 say " " color (colorvar)
204 @ 1,2 say " " color (colorvar)
205
206 @ 2,0 say " " color w/n
207
208 @ 3,0 say " " color w/n
209
210 @ 4,0 say " " color (colorvar)
211 --case substr(instring,curlet,1) = "n"
212 define window ("letter" + alltrim(str(curlet))) ;
213 from startx,curenty to startx + 5,curenty + 5 ;
214 none noshadow
215 activate window ("letter" + alltrim(str(curlet)))
216 curenty = curenty + 7
217 @ 1,0 say " " color w/n
218 @ 1,2 say " " color w/n
219 @ 1,4 say " " color w/n
220
221 @ 2,0 say " " color w/n
222
223 @ 3,0 say " " color w/n
224
225 @ 4,0 say " " color (colorvar)
226 --case substr(instring,curlet,1) = "m"
227 define window ("letter" + alltrim(str(curlet))) ;
228 from startx,curenty to startx + 5,curenty + 5 ;
229 none noshadow
230 activate window ("letter" + alltrim(str(curlet)))
231 curenty = curenty + 10
232 @ 1,0 say " " color w/n
233 @ 1,2 say " " color w/n
234 @ 1,4 say " " color w/n
235 @ 1,5 say " " color w/n
236 @ 1,7 say " " color w/n
237
238 @ 2,0 say " " color w/n
239
240 @ 3,0 say " " color w/n
241
242 @ 4,0 say " " color (colorvar)
243 --case substr(instring,curlet,1) = "f"
244 define window ("letter" + alltrim(str(curlet))) ;
245 from startx,curenty to startx + 5,curenty + 6 ;
246 none noshadow
247 activate window ("letter" + alltrim(str(curlet)))
248 curenty = curenty + 7
249 @ 0,0 say " " color w/n
250 @ 0,3 say " " color (colorvar)
251 @ 0,5 say " " color (colorvar)
252
253 @ 1,0 say " " color w/n
254 @ 1,3 say " " color (colorvar)
255 @ 1,4 say " " color (colorvar)
256
257 @ 2,1 say " " color w/n
258
259 @ 3,1 say " " color w/n
260
261 @ 4,0 say " " color (colorvar)
262 --case substr(instring,curlet,1) = "l"
263 define window ("letter" + alltrim(str(curlet))) ;
264 from startx,curenty to startx + 5,curenty + 5 ;

```

```

265 none noshadow
266 activate window ("letter" + alltrim(str(curllet)))
267 currenly = currenly + 4
268 @ 0,0 say " " color (colorvar)
269 @ 0,2 say " " color (colorvar)
270
271 @ 1,0 say " " color w/n
272
273 @ 2,0 say " " color w/n
274
275 @ 3,0 say " " color w/n
276
277 @ 4,0 say " " color (colorvar)
278
279 -case substr(instring,curlet,1) = "q"
280 define window ("letter" + alltrim(str(curllet))) ;
281 from startx,currenly to startx + 5,currenly + 5 ;
282 none noshadow
283 activate window ("letter" + alltrim(str(curllet)))
284 currenly = currenly + 8
285 @ 0,0 say " " color (colorvar)
286 @ 0,2 say " " color (colorvar)
287
288 @ 1,0 say " " color w/n
289 @ 1,5 say " " color (colorvar)
290
291 @ 2,0 say " " color w/n
292 @ 2,4 say " " color (colorvar)
293
294 @ 3,0 say " " color w/n
295 @ 3,3 say " " color (colorvar)
296
297 @ 4,0 say " " color (colorvar)
298
299 -case substr(instring,curlet,1) = "u"
300 define window ("letter" + alltrim(str(curllet))) ;
301 from startx,currenly to startx + 5,currenly + 6 ;
302 none noshadow
303 activate window ("letter" + alltrim(str(curllet)))
304 currenly = currenly + 7
305 @ 1,0 say " " color (colorvar)
306 @ 1,2 say " " color (colorvar)
307 @ 1,3 say " " color (colorvar)
308 @ 1,5 say " " color (colorvar)
309
310 @ 2,0 say " " color w/n
311
312 @ 3,0 say " " color w/n
313 @ 3,3 say " " color w/n
314 @ 3,4 say " " color w/n
315
316 @ 4,0 say " " color (colorvar)
317
318 -case substr(instring,curlet,1) = "p"
319 define window ("letter" + alltrim(str(curllet))) ;
320 from startx,currenly to startx + 5,currenly + 6 ;
321 none noshadow
322 activate window ("letter" + alltrim(str(curllet)))
323 currenly = currenly + 7
324 @ 0,0 say " " color (colorvar)
325 @ 0,2 say " " color (colorvar)
326
327 @ 1,0 say " " color w/n
328 @ 1,4 say " " color (colorvar)
329
330 @ 2,0 say " " color w/n
331 @ 2,3 say " " color (colorvar)
332 @ 2,5 say " " color (colorvar)

```

```

331
332
333
334 @ 3,0 say " " color w/n
335
336 @ 4,0 say " " color (colorvar)
337
338 -case substr(instring,curlet,1) = "c"
339 define window ("letter" + alltrim(str(curllet))) ;
340 from startx,currenly to startx + 5,currenly + 5 ;
341 none noshadow
342 activate window ("letter" + alltrim(str(curllet)))
343 currenly = currenly + 6
344 @ 1,0 say " " color w/n
345 @ 1,2 say " " color (colorvar)
346 @ 1,4 say " " color (colorvar)
347
348 @ 2,0 say " " color w/n
349 @ 2,3 say " " color (colorvar)
350
351 @ 3,0 say " " color w/n
352 @ 3,2 say " " color (colorvar)
353 @ 3,4 say " " color (colorvar)
354
355 @ 4,0 say " " color (colorvar)
356
357 -case substr(instring,curlet,1) = "g"
358 define window ("letter" + alltrim(str(curllet))) ;
359 from startx,currenly to startx + 5,currenly + 5 ;
360 none noshadow
361 activate window ("letter" + alltrim(str(curllet)))
362 currenly = currenly + 7
363 @ 0,0 say " " color (colorvar)
364 @ 0,5 say " " color (colorvar)
365
366 @ 1,0 say " " color w/n
367 @ 1,2 say " " color w/n
368
369 @ 2,0 say " " color w/n
370 @ 2,2 say " " color (colorvar)
371 @ 2,3 say " " color w/n
372
373 @ 3,0 say " " color (colorvar)
374 @ 3,5 say " " color (colorvar)
375
376 @ 4,0 say " " color (colorvar)
377
378 -case substr(instring,curlet,1) = "h"
379 define window ("letter" + alltrim(str(curllet))) ;
380 from startx,currenly to startx + 5,currenly + 5 ;
381 none noshadow
382 activate window ("letter" + alltrim(str(curllet)))
383 currenly = currenly + 7
384 @ 1,0 say " " color w/n
385 @ 1,2 say " " color (colorvar)
386 @ 1,3 say " " color (colorvar)
387 @ 1,5 say " " color w/n
388
389 @ 2,0 say " " color w/n
390 @ 2,2 say " " color (colorvar)
391 @ 2,3 say " " color w/n
392
393 @ 3,0 say " " color (colorvar)
394 @ 3,5 say " " color (colorvar)
395
396 @ 4,0 say " " color (colorvar)
397 @ 4,3 say " " color w/n
398
399 @ 5,0 say " " color (colorvar)
400
401 -case substr(instring,curlet,1) = "g"

```

```

397 define window ("letter" + alltrim(str(curlet))) ;
398   from startx, currenty to startx + 5, currenty + 6 ;
399   none noshadow
400   activate window ("letter" + alltrim(str(curlet)))
401   currenty = currenty + 7
402   @ 1,0 say " " color w/n
403   @ 1,2 say " " color w/n
404   @ 1,3 say " " color (colorvar)
405   @ 1,5 say " " color (colorvar)
406   @ 2,0 say " " color w/n
407   @ 2,2 say " " color (colorvar)
408   @ 2,3 say " " color w/n
409   @ 3,0 say " " color (colorvar)
410   @ 3,5 say " " color (colorvar)
411   @ 4,0 say " " color w/n
412   @ 4,2 say " " color w/n
413   @ 4,3 say " " color w/n
414   @ 4,4 say " " color w/n
415   @ 5,0 say " " color (colorvar)
416   -case substr(instring,curlet,1) = "h"
417   define window ("letter" + alltrim(str(curlet))) ;
418   from startx, currenty to startx + 5, currenty + 6 ;
419   none noshadow
420   activate window ("letter" + alltrim(str(curlet)))
421   currenty = currenty + 7
422   @ 0,0 say " " color w/n
423   @ 0,2 say " " color (colorvar)
424   @ 1,0 say " " color w/n
425   @ 1,4 say " " color w/n
426   @ 2,0 say " " color w/n
427   @ 3,0 say " " color w/n
428   @ 4,0 say " " color (colorvar)
429   -case substr(instring,curlet,1) = "j"
430   define window ("letter" + alltrim(str(curlet))) ;
431   from startx, currenty to startx + 5, currenty + 5 ;
432   none noshadow
433   activate window ("letter" + alltrim(str(curlet)))
434   currenty = currenty + 5
435   @ 0,0 say " " color w/n
436   @ 0,2 say " " color w/n
437   @ 0,3 say " " color (colorvar)
438   @ 1,0 say " " color (colorvar)
439   @ 1,3 say " " color (colorvar)
440   @ 2,1 say " " color w/n
441   @ 3,1 say " " color w/n
442   @ 4,0 say " " color w/n
443   @ 4,3 say " " color (colorvar)
444   -case substr(instring,curlet,1) = "p"
445   define window ("letter" + alltrim(str(curlet))) ;
446   from startx, currenty to startx + 5, currenty + 6 ;
447   none noshadow
448   activate window ("letter" + alltrim(str(curlet)))
449   currenty = currenty + 7
450   @ 4,0 say " " color w/n
451   @ 4,1 say " " color w/n
452   @ 4,3 say " " color (colorvar)
453   @ 4,5 say " " color (colorvar)
454   -case substr(instring,curlet,1) = "p"
455   define window ("letter" + alltrim(str(curlet))) ;
456   from startx, currenty to startx + 5, currenty + 6 ;
457   none noshadow
458   activate window ("letter" + alltrim(str(curlet)))
459   currenty = currenty + 7
460   @ 4,0 say " " color w/n
461   @ 4,1 say " " color w/n
462   @ 4,3 say " " color w/n

```

```

463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528

```

```

@ 1,0 say " " color w/n
@ 2,0 say " " color w/n
@ 2,3 say " " color (colorvar)
@ 2,5 say " " color (colorvar)
@ 3,0 say " " color w/n
@ 4,0 say " " color w/n
@ 4,3 say " " color (colorvar)
@ 5,0 say " " color (colorvar)
-case substr(instring,curlet,1) = "h"
define window ("letter" + alltrim(str(curlet))) ;
from startx, currenty to startx + 5, currenty + 6 ;
none noshadow
activate window ("letter" + alltrim(str(curlet)))
currenty = currenty + 9
@ 1,0 say " " color (colorvar)
@ 1,2 say " " color (colorvar)
@ 1,3 say " " color (colorvar)
@ 1,4 say " " color (colorvar)
@ 1,5 say " " color (colorvar)
@ 1,7 say " " color (colorvar)
@ 2,0 say " " color w/n
@ 2,2 say " " color w/n
@ 3,1 say " " color w/n
@ 4,0 say " " color (colorvar)
-case substr(instring,curlet,1) = "j"
define window ("letter" + alltrim(str(curlet))) ;
from startx, currenty to startx + 5, currenty + 7 ;
none noshadow
activate window ("letter" + alltrim(str(curlet)))
currenty = currenty + 8
@ 1,0 say " " color w/n
@ 1,5 say " " color (colorvar)
@ 1,6 say " " color (colorvar)
@ 2,0 say " " color w/n
@ 2,4 say " " color w/n
@ 2,5 say " " color (colorvar)
@ 3,0 say " " color w/n
@ 3,3 say " " color (colorvar)
@ 3,4 say " " color w/n
@ 3,6 say " " color (colorvar)
@ 4,0 say " " color (colorvar)
-case substr(instring,curlet,1) = "p"
define window ("letter" + alltrim(str(curlet))) ;
from startx, currenty to startx + 5, currenty + 6 ;
none noshadow
activate window ("letter" + alltrim(str(curlet)))
currenty = currenty + 7
@ 1,0 say " " color (colorvar)
@ 1,2 say " " color (colorvar)
@ 1,3 say " " color (colorvar)
@ 1,5 say " " color (colorvar)
@ 2,0 say " " color w/n
@ 3,0 say " " color w/n
@ 3,4 say " " color w/n

```

```

529 @ 4,0 say " " color w/n
530 @ 4,3 say " " color w/n
531
532 @ 5,0 say " " color (colorvar)
533
534 -case substr(instring,curlet,1) = "y"
535   define window ("letter" + alltrim(str(curlet))) ;
536   from startx,curlet to startx + 5,curlet + 6 ;
537   none noshadow
538   activate window ("letter" + alltrim(str(curlet)))
539   curlet = curlet + 7
540 @ 1,0 say " " color (colorvar)
541 @ 1,2 say " " color (colorvar)
542 @ 1,3 say " " color (colorvar)
543 @ 1,5 say " " color (colorvar)
544
545 @ 2,0 say " " color (colorvar)
546 @ 2,1 say " " color w/n
547
548 @ 3,0 say " " color w/n
549 @ 3,3 say " " color w/n
550
551 @ 4,0 say " " color (colorvar)
552 -case substr(instring,curlet,1) = "y"
553   define window ("letter" + alltrim(str(curlet))) ;
554   from startx,curlet to startx + 5,curlet + 6 ;
555   none noshadow
556   activate window ("letter" + alltrim(str(curlet)))
557   curlet = curlet + 6
558 @ 0,0 say " " color (colorvar)
559 @ 0,4 say " " color (colorvar)
560
561 @ 1,0 say " " color (colorvar)
562 @ 1,3 say " " color (colorvar)
563
564 @ 2,0 say " " color (colorvar)
565 @ 2,2 say " " color (colorvar)
566
567 @ 3,0 say " " color (colorvar)
568 @ 3,1 say " " color (colorvar)
569
570 @ 4,0 say " " color (colorvar)
571 -case substr(instring,curlet,1) = "z"
572   define window ("letter" + alltrim(str(curlet))) ;
573   from startx,curlet to startx + 5,curlet + 6 ;
574   none noshadow
575   activate window ("letter" + alltrim(str(curlet)))
576   curlet = curlet + 7
577 @ 1,0 say " " color w/n
578 @ 1,1 say " " color (colorvar)
579 @ 1,5 say " " color (colorvar)
580
581 @ 2,0 say " " color w/n
582 @ 2,3 say " " color (colorvar)
583 @ 2,5 say " " color (colorvar)
584
585 @ 3,0 say " " color w/n
586 @ 3,4 say " " color w/n
587
588 @ 4,0 say " " color (colorvar)
589 -case substr(instring,curlet,1) = "a"
590   startx = startx + 6
591   curlet = startx
592 -case substr(instring,curlet,1) = "a"
593   define window ("letter" + alltrim(str(curlet))) ;
594   from startx,curlet to startx + 5,curlet + 7 ;

```

```

595
596 none noshadow
597 activate window ("letter" + alltrim(str(curlet)))
598 curlet = curlet + 9
599 @ 0,2 say " " color w/n
600 @ 0,5 say " " color (colorvar)
601
602 @ 1,1 say " " color w/n
603 @ 1,6 say " " color (colorvar)
604
605 @ 2,0 say " " color w/n
606 @ 2,1 say " " color w/n
607
608 @ 3,0 say " " color w/n
609 @ 3,3 say " " color (colorvar)
610 @ 3,5 say " " color w/n
611
612 @ 4,0 say " " color (colorvar)
613 -case substr(instring,curlet,1) = "g"
614   define window ("letter" + alltrim(str(curlet))) ;
615   from startx,curlet to startx + 5,curlet + 6 ;
616   none noshadow
617   activate window ("letter" + alltrim(str(curlet)))
618   curlet = curlet + 8
619 @ 0,0 say " " color w/n
620 @ 0,4 say " " color w/n
621
622 @ 1,0 say " " color w/n
623 @ 1,3 say " " color w/n
624 @ 1,5 say " " color w/n
625
626 @ 2,0 say " " color w/n
627 @ 2,3 say " " color (colorvar)
628 @ 2,4 say " " color w/n
629 @ 2,6 say " " color (colorvar)
630
631 @ 3,0 say " " color w/n
632 @ 3,3 say " " color w/n
633 @ 3,5 say " " color w/n
634
635 @ 4,0 say " " color (colorvar)
636 -case substr(instring,curlet,1) = "c"
637   define window ("letter" + alltrim(str(curlet))) ;
638   from startx,curlet to startx + 5,curlet + 8 ;
639   none noshadow
640   activate window ("letter" + alltrim(str(curlet)))
641   curlet = curlet + 9
642 @ 0,0 say " " color w/n
643 @ 0,3 say " " color w/n
644 @ 0,6 say " " color w/n
645
646 @ 1,0 say " " color w/n
647 @ 1,3 say " " color (colorvar)
648 @ 1,5 say " " color (colorvar)
649 @ 1,7 say " " color (colorvar)
650
651 @ 2,0 say " " color w/n
652 @ 2,3 say " " color w/n
653 @ 3,0 say " " color w/n
654 @ 3,6 say " " color w/n
655
656 @ 4,0 say " " color (colorvar)
657 -case substr(instring,curlet,1) = "w"
658   define window ("letter" + alltrim(str(curlet))) ;
659   from startx,curlet to startx + 5,curlet + 6 ;
660   none noshadow

```



```

661 activate window ("letter" + alltrim(str(curlet)))
662 currenty = currenty + 8
663 @ 0,0 say " " color w/n
664 @ 0,4 say " " color w/n
665 @ 1,0 say " " color w/n
666 @ 1,3 say " " color (colorvar)
667 @ 1,6 say " " color w/n
668 @ 2,0 say " " color w/n
669 @ 2,3 say " " color (colorvar)
670 @ 2,6 say " " color w/n
671 @ 3,0 say " " color w/n
672 @ 3,6 say " " color (colorvar)
673 @ 4,0 say " " color (colorvar)
674 case substr(instring,curlet,1) = "H"
675 define window ("letter" + alltrim(str(curlet))) ;
676 from startx,currenty to startx + 5,currenty + 6 ;
677 none noshadow
678 activate window ("letter" + alltrim(str(curlet)))
679 currenty = currenty + 7
680 @ 0,0 say " " color w/n
681 @ 0,5 say " " color (colorvar)
682 @ 1,0 say " " color w/n
683 @ 1,3 say " " color w/n
684 @ 2,0 say " " color w/n
685 @ 2,3 say " " color (colorvar)
686 @ 3,0 say " " color w/n
687 @ 3,3 say " " color w/n
688 @ 4,0 say " " color (colorvar)
689 case substr(instring,curlet,1) = "M"
690 define window ("letter" + alltrim(str(curlet))) ;
691 from startx,currenty to startx + 5,currenty + 6 ;
692 none noshadow
693 activate window ("letter" + alltrim(str(curlet)))
694 currenty = currenty + 7
695 @ 0,0 say " " color w/n
696 @ 0,5 say " " color (colorvar)
697 @ 1,0 say " " color w/n
698 @ 1,3 say " " color w/n
699 @ 2,0 say " " color w/n
700 @ 2,3 say " " color (colorvar)
701 @ 3,0 say " " color w/n
702 @ 3,3 say " " color w/n
703 @ 4,0 say " " color (colorvar)
704 case substr(instring,curlet,1) = "G"
705 define window ("letter" + alltrim(str(curlet))) ;
706 from startx,currenty to startx + 5,currenty + 6 ;
707 none noshadow
708 activate window ("letter" + alltrim(str(curlet)))
709 currenty = currenty + 7
710 @ 0,0 say " " color w/n
711 @ 0,5 say " " color (colorvar)
712 @ 1,0 say " " color w/n
713 @ 1,3 say " " color w/n
714 @ 2,0 say " " color w/n
715 @ 2,3 say " " color (colorvar)
716 @ 3,0 say " " color w/n
717 @ 3,3 say " " color w/n
718 @ 4,0 say " " color (colorvar)
719 case substr(instring,curlet,1) = "J"
720 define window ("letter" + alltrim(str(curlet))) ;
721 from startx,currenty to startx + 5,currenty + 6 ;
722 none noshadow
723 activate window ("letter" + alltrim(str(curlet)))
724 currenty = currenty + 9
725 @ 0,0 say " " color w/n
726 @ 0,6 say " " color w/n
727 @ 1,0 say " " color w/n
728 @ 1,3 say " " color w/n
729 @ 2,0 say " " color w/n
730 @ 2,3 say " " color (colorvar)
731 @ 3,0 say " " color w/n
732 @ 3,6 say " " color (colorvar)
733 @ 4,0 say " " color w/n
734 case substr(instring,curlet,1) = "H"
735 define window ("letter" + alltrim(str(curlet))) ;
736 from startx,currenty to startx + 5,currenty + 7 ;
737 none noshadow
738 activate window ("letter" + alltrim(str(curlet)))
739 currenty = currenty + 8
740 @ 0,0 say " " color w/n
741 @ 0,2 say " " color (colorvar)
742 @ 0,4 say " " color (colorvar)
743 @ 0,6 say " " color (colorvar)
744 @ 1,0 say " " color w/n
745 @ 1,3 say " " color (colorvar)
746 @ 1,6 say " " color (colorvar)
747 @ 2,0 say " " color w/n
748 @ 2,3 say " " color w/n
749 @ 2,4 say " " color w/n
750 @ 3,0 say " " color w/n
751 @ 3,3 say " " color w/n
752 @ 3,4 say " " color w/n
753 @ 4,0 say " " color (colorvar)
754 case substr(instring,curlet,1) = "J"
755 define window ("letter" + alltrim(str(curlet))) ;
756 from startx,currenty to startx + 5,currenty + 3 ;
757 none noshadow
758 activate window ("letter" + alltrim(str(curlet)))
759 currenty = currenty + 4
760 @ 0,0 say " " color (colorvar)
761 @ 0,2 say " " color (colorvar)
762 @ 1,0 say " " color w/n
763 @ 2,0 say " " color w/n
764 @ 2,3 say " " color w/n
765 @ 3,0 say " " color w/n
766 @ 3,3 say " " color w/n
767 @ 4,0 say " " color (colorvar)
768 case substr(instring,curlet,1) = "J"
769 define window ("letter" + alltrim(str(curlet))) ;
770 from startx,currenty to startx + 5,currenty + 6 ;
771 none noshadow
772 activate window ("letter" + alltrim(str(curlet)))
773 currenty = currenty + 7
774 @ 0,0 say " " color w/n
775 @ 0,2 say " " color (colorvar)
776 @ 1,0 say " " color w/n
777 @ 2,0 say " " color w/n
778 @ 2,3 say " " color w/n
779 @ 3,0 say " " color w/n
780 @ 4,0 say " " color (colorvar)
781 case substr(instring,curlet,1) = "G"
782 define window ("letter" + alltrim(str(curlet))) ;
783 from startx,currenty to startx + 5,currenty + 6 ;
784 none noshadow
785 activate window ("letter" + alltrim(str(curlet)))
786 currenty = currenty + 9
787 @ 0,0 say " " color w/n
788 @ 0,3 say " " color w/n
789 @ 0,6 say " " color w/n
790 @ 1,0 say " " color w/n
791 @ 1,3 say " " color (colorvar)
792

```

```

727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792

```

```

793 @ 4,0 say " " color (colorvar)
794 -case substr(instrng,curlet,1) = "q"
795 define window ("letter" + alltrim(str(curlet))) ;
796 from startx,curlet to startx + 5,curlet + 7 ;
797 none noshadow
798 activate window ("letter" + alltrim(str(curlet)))
799 currenty = currenty + 9
800 @ 0,0 say " " color (colorvar)
801 @ 0,2 say " " color (colorvar)
802 @ 0,3 say " " color w/n
803 @ 0,5 say " " color (colorvar)
804 @ 0,6 say " " color (colorvar)
805
806 @ 1,0 say " " color w/n
807 @ 1,5 say " " color (colorvar)
808
809 @ 2,0 say " " color w/n
810 @ 2,4 say " " color w/n
811
812 @ 3,0 say " " color w/n
813 @ 3,3 say " " color w/n
814
815 @ 4,0 say " " color (colorvar)
816 -case substr(instrng,curlet,1) = "l"
817 define window ("letter" + alltrim(str(curlet))) ;
818 from startx,curlet to startx + 5,curlet + 6 ;
819 none noshadow
820 activate window ("letter" + alltrim(str(curlet)))
821 currenty = currenty + 7
822 @ 0,0 say " " color (colorvar)
823 @ 0,2 say " " color (colorvar)
824
825 @ 1,0 say " " color w/n
826
827 @ 2,0 say " " color w/n
828
829 @ 3,0 say " " color w/n
830 @ 3,3 say " " color w/n
831
832 @ 4,0 say " " color (colorvar)
833 -case substr(instrng,curlet,1) = "p"
834 define window ("letter" + alltrim(str(curlet))) ;
835 from startx,curlet to startx + 5,curlet + 10 ;
836 none noshadow
837 activate window ("letter" + alltrim(str(curlet)))
838 currenty = currenty + 11
839 @ 0,0 say " " color (colorvar)
840 @ 0,9 say " " color (colorvar)
841
842 @ 1,0 say " " color w/n
843 @ 1,4 say " " color (colorvar)
844 @ 1,9 say " " color (colorvar)
845
846 @ 2,0 say " " color w/n
847
848 @ 3,0 say " " color w/n
849
850 @ 3,3 say " " color (colorvar)
851 @ 3,5 say " " color (colorvar)
852 @ 3,7 say " " color w/n
853
854 @ 4,0 say " " color (colorvar)
855 -case substr(instrng,curlet,1) = "h"
856 define window ("letter" + alltrim(str(curlet))) ;
857 from startx,curlet to startx + 5,curlet + 6 ;
858

```

```

859 none noshadow
860 activate window ("letter" + alltrim(str(curlet)))
861 currenty = currenty + 9
862 @ 0,0 say " " color (colorvar)
863 @ 0,7 say " " color (colorvar)
864
865 @ 1,0 say " " color w/n
866 @ 1,4 say " " color (colorvar)
867 @ 1,7 say " " color (colorvar)
868
869 @ 2,0 say " " color w/n
870 @ 2,3 say " " color w/n
871 @ 2,4 say " " color w/n
872
873 @ 3,0 say " " color w/n
874 @ 3,3 say " " color w/n
875 @ 3,7 say " " color w/n
876
877 @ 4,0 say " " color (colorvar)
878 -case substr(instrng,curlet,1) = "q"
879 define window ("letter" + alltrim(str(curlet))) ;
880 from startx,curlet to startx + 7,curlet + 7 ;
881 none noshadow
882 activate window ("letter" + alltrim(str(curlet)))
883 currenty = currenty + 9
884 @ 0,0 say " " color w/n
885 @ 0,3 say " " color w/n
886 @ 0,4 say " " color w/n
887
888 @ 1,0 say " " color w/n
889 @ 1,3 say " " color (colorvar)
890 @ 1,5 say " " color (colorvar)
891 @ 1,7 say " " color (colorvar)
892
893 @ 2,0 say " " color w/n
894 @ 2,3 say " " color (colorvar)
895 @ 2,7 say " " color (colorvar)
896
897 @ 3,0 say " " color w/n
898 @ 3,5 say " " color w/n
899 @ 3,7 say " " color (colorvar)
900
901 @ 4,0 say " " color (colorvar)
902 -case substr(instrng,curlet,1) = "p"
903 define window ("letter" + alltrim(str(curlet))) ;
904 from startx,curlet to startx + 5,curlet + 6 ;
905 none noshadow
906 activate window ("letter" + alltrim(str(curlet)))
907 currenty = currenty + 8
908 @ 0,0 say " " color w/n
909 @ 0,3 say " " color w/n
910
911 @ 1,0 say " " color w/n
912 @ 1,3 say " " color (colorvar)
913 @ 1,6 say " " color (colorvar)
914
915 @ 2,0 say " " color w/n
916 @ 2,6 say " " color (colorvar)
917 @ 2,6 say " " color (colorvar)
918 @ 3,0 say " " color w/n
919
920 @ 4,0 say " " color (colorvar)
921 -case substr(instrng,curlet,1) = "h"
922 define window ("letter" + alltrim(str(curlet))) ;
923 from startx,curlet to startx + 5,curlet + 6 ;
924

```

```

925 none noshadow
926 activate window ("letter" + alltrim(str(curlet)))
927 currenly = currenly + 9
928 @ 0,0 say " " color w/n
929 @ 0,3 say " " color w/n
930 @ 0,4 say " " color w/n
931
932 @ 1,0 say " " color w/n
933 @ 1,3 say " " color (colorvar)
934 @ 1,5 say " " color (colorvar)
935 @ 1,7 say " " color (colorvar)
936
937 @ 2,0 say " " color w/n
938 @ 2,3 say " " color (colorvar)
939 @ 2,7 say " " color (colorvar)
940
941 @ 3,0 say " " color (colorvar)
942 @ 3,6 say " " color (colorvar)
943
944 @ 4,0 say " " color (colorvar)
945 @ 4,6 say " " color (colorvar)
946 @ 4,7 say " " color (colorvar)
947
948 define window ("letter" + alltrim(str(curlet))) ;
949 from startx,currenly to startx + 5,currenly + 7 ;
950 none noshadow
951 activate window ("letter" + alltrim(str(curlet)))
952 currenly = currenly + 8
953 @ 0,0 say " " color w/n
954 @ 0,3 say " " color w/n
955
956 @ 1,0 say " " color w/n
957 @ 1,3 say " " color (colorvar)
958 @ 1,6 say " " color (colorvar)
959
960 @ 2,0 say " " color w/n
961 @ 2,6 say " " color (colorvar)
962
963 @ 3,0 say " " color w/n
964 @ 3,3 say " " color (colorvar)
965 @ 3,6 say " " color (colorvar)
966
967 @ 4,0 say " " color (colorvar)
968
969 define window ("letter" + alltrim(str(curlet))) ;
970 from startx,currenly to startx + 5,currenly + 7 ;
971 none noshadow
972 activate window ("letter" + alltrim(str(curlet)))
973 currenly = currenly + 8
974 @ 0,0 say " " color w/n
975 @ 0,2 say " " color w/n
976 @ 0,5 say " " color w/n
977
978 @ 1,0 say " " color (colorvar)
979 @ 1,5 say " " color (colorvar)
980
981 @ 2,0 say " " color (colorvar)
982 @ 2,6 say " " color (colorvar)
983
984 @ 3,0 say " " color w/n
985 @ 3,5 say " " color w/n
986
987 @ 4,0 say " " color (colorvar)
988
989 define window ("letter" + alltrim(str(curlet))) ;
990 from startx,currenly to startx + 5,currenly + 7 ;

```

```

991 none noshadow
992 activate window ("letter" + alltrim(str(curlet)))
993 currenly = currenly + 8
994 @ 0,0 say " " color w/n
995 @ 0,1 say " " color (colorvar)
996 @ 0,6 say " " color (colorvar)
997
998 @ 1,2 say " " color w/n
999 @ 2,2 say " " color w/n
1000 @ 3,2 say " " color w/n
1001
1002 @ 4,0 say " " color (colorvar)
1003
1004 define window ("letter" + alltrim(str(curlet))) ;
1005 from startx,currenly to startx + 5,currenly + 7 ;
1006 none noshadow
1007 activate window ("letter" + alltrim(str(curlet)))
1008 currenly = currenly + 8
1009 @ 0,0 say " " color (colorvar)
1010 @ 0,2 say " " color (colorvar)
1011 @ 0,4 say " " color (colorvar)
1012 @ 0,6 say " " color (colorvar)
1013
1014 @ 1,0 say " " color w/n
1015 @ 1,3 say " " color (colorvar)
1016 @ 1,6 say " " color (colorvar)
1017
1018 @ 2,0 say " " color w/n
1019 @ 2,3 say " " color w/n
1020 @ 2,4 say " " color w/n
1021
1022 @ 3,0 say " " color w/n
1023 @ 3,2 say " " color w/n
1024 @ 3,3 say " " color w/n
1025 @ 3,4 say " " color w/n
1026
1027 @ 4,0 say " " color (colorvar)
1028
1029 define window ("letter" + alltrim(str(curlet))) ;
1030 from startx,currenly to startx + 5,currenly + 7 ;
1031 none noshadow
1032 activate window ("letter" + alltrim(str(curlet)))
1033 currenly = currenly + 9
1034 @ 0,0 say " " color (colorvar)
1035 @ 0,2 say " " color (colorvar)
1036 @ 0,3 say " " color (colorvar)
1037 @ 0,7 say " " color (colorvar)
1038
1039 @ 1,0 say " " color (colorvar)
1040 @ 1,3 say " " color (colorvar)
1041 @ 1,4 say " " color w/n
1042 @ 1,7 say " " color (colorvar)
1043
1044 @ 2,0 say " " color w/n
1045 @ 2,3 say " " color w/n
1046 @ 3,0 say " " color w/n
1047 @ 3,4 say " " color w/n
1048
1049 @ 4,0 say " " color (colorvar)
1050
1051 define window ("letter" + alltrim(str(curlet))) ;
1052 from startx,currenly to startx + 5,currenly + 10 ;
1053 none noshadow
1054 activate window ("letter" + alltrim(str(curlet)))
1055 currenly = currenly + 10
1056

```

```

1057 activate window ("letter" + alltrim(str(curlet)))
1058 cumenty = cumenty + 12
1059 @ 0,0 say " " color (colorvar)
1060 @ 0,2 say " " color (colorvar)
1061 @ 0,4 say " " color (colorvar)
1062 @ 0,6 say " " color (colorvar)
1063 @ 0,8 say " " color (colorvar)
1064 @ 0,10 say " " color (colorvar)
1065 @ 1,0 say " " color w/n
1066 @ 1,2 say " " color w/n
1067 @ 1,3 say " " color w/n
1068 @ 1,6 say " " color (colorvar)
1069 @ 1,7 say " " color w/n
1070 @ 1,9 say " " color w/n
1071 @ 2,1 say " " color w/n
1072 @ 3,2 say " " color w/n
1073 @ 3,6 say " " color w/n
1074 @ 3,9 say " " color (colorvar)
1075 @ 4,0 say " " color (colorvar)
1076 define window ("letter" + alltrim(str(curlet))) ;
1077 from startx,cumenty to startx + 5,cumenty + 8 ;
1078 none noshadow
1079 activate window ("letter" + alltrim(str(curlet)))
1080 cumenty = cumenty + 9
1081 @ 0,0 say " " color (colorvar)
1082 @ 0,2 say " " color (colorvar)
1083 @ 0,3 say " " color (colorvar)
1084 @ 0,7 say " " color (colorvar)
1085 @ 1,0 say " " color w/n
1086 @ 1,5 say " " color w/n
1087 @ 1,6 say " " color (colorvar)
1088 @ 2,0 say " " color w/n
1089 @ 2,3 say " " color w/n
1090 @ 3,0 say " " color w/n
1091 @ 3,3 say " " color (colorvar)
1092 @ 3,4 say " " color (colorvar)
1093 @ 3,7 say " " color (colorvar)
1094 @ 4,0 say " " color (colorvar)
1095 @ 4,0 say " " color (colorvar)
1096 @ 4,0 say " " color (colorvar)
1097 @ 4,0 say " " color (colorvar)
1098 @ 4,0 say " " color (colorvar)
1099 @ 4,0 say " " color (colorvar)
1100 @ 4,0 say " " color (colorvar)
1101 @ 4,0 say " " color (colorvar)
1102 @ 4,0 say " " color (colorvar)
1103 @ 4,0 say " " color (colorvar)
1104 @ 4,0 say " " color (colorvar)
1105 @ 4,0 say " " color (colorvar)
1106 @ 4,0 say " " color (colorvar)
1107 @ 4,0 say " " color (colorvar)
1108 @ 4,0 say " " color (colorvar)
1109 @ 4,0 say " " color (colorvar)
1110 @ 4,0 say " " color (colorvar)
1111 @ 4,0 say " " color (colorvar)
1112 @ 4,0 say " " color (colorvar)
1113 @ 4,0 say " " color (colorvar)
1114 @ 4,0 say " " color (colorvar)
1115 @ 4,0 say " " color (colorvar)
1116 @ 4,0 say " " color (colorvar)
1117 @ 4,0 say " " color (colorvar)
1118 @ 4,0 say " " color (colorvar)
1119 @ 4,0 say " " color (colorvar)
1120 @ 4,0 say " " color (colorvar)
1121 @ 4,0 say " " color (colorvar)
1122 @ 4,0 say " " color (colorvar)

```

```

1123 @ 3,3 say " " color w/n
1124 @ 4,0 say " " color (colorvar)
1125 @ 4,0 say " " color (colorvar)
1126 @ 4,0 say " " color (colorvar)
1127 @ 4,0 say " " color (colorvar)
1128 @ 4,0 say " " color (colorvar)
1129 @ 4,0 say " " color (colorvar)
1130 @ 4,0 say " " color (colorvar)
1131 @ 4,0 say " " color (colorvar)
1132 @ 4,0 say " " color (colorvar)
1133 @ 4,0 say " " color (colorvar)
1134 @ 4,0 say " " color (colorvar)
1135 @ 4,0 say " " color (colorvar)
1136 @ 4,0 say " " color (colorvar)
1137 @ 4,0 say " " color (colorvar)
1138 @ 4,0 say " " color (colorvar)
1139 @ 4,0 say " " color (colorvar)
1140 @ 4,0 say " " color (colorvar)
1141 @ 4,0 say " " color (colorvar)
1142 @ 4,0 say " " color (colorvar)
1143 @ 4,0 say " " color (colorvar)
1144 @ 4,0 say " " color (colorvar)
1145 @ 4,0 say " " color (colorvar)
1146 @ 4,0 say " " color (colorvar)
1147 @ 4,0 say " " color (colorvar)
1148 @ 4,0 say " " color (colorvar)
1149 @ 4,0 say " " color (colorvar)
1150 @ 4,0 say " " color (colorvar)
1151 @ 4,0 say " " color (colorvar)
1152 @ 4,0 say " " color (colorvar)
1153 @ 4,0 say " " color (colorvar)
1154 @ 4,0 say " " color (colorvar)
1155 @ 4,0 say " " color (colorvar)
1156 @ 4,0 say " " color (colorvar)
1157 @ 4,0 say " " color (colorvar)
1158 @ 4,0 say " " color (colorvar)
1159 @ 4,0 say " " color (colorvar)
1160 @ 4,0 say " " color (colorvar)
1161 @ 4,0 say " " color (colorvar)
1162 @ 4,0 say " " color (colorvar)
1163 @ 4,0 say " " color (colorvar)
1164 @ 4,0 say " " color (colorvar)
1165 @ 4,0 say " " color (colorvar)
1166 @ 4,0 say " " color (colorvar)
1167 @ 4,0 say " " color (colorvar)
1168 @ 4,0 say " " color (colorvar)
1169 @ 4,0 say " " color (colorvar)
1170 @ 4,0 say " " color (colorvar)
1171 @ 4,0 say " " color (colorvar)
1172 @ 4,0 say " " color (colorvar)
1173 @ 4,0 say " " color (colorvar)
1174 @ 4,0 say " " color (colorvar)
1175 @ 4,0 say " " color (colorvar)
1176 @ 4,0 say " " color (colorvar)
1177 @ 4,0 say " " color (colorvar)
1178 @ 4,0 say " " color (colorvar)
1179 @ 4,0 say " " color (colorvar)
1180 @ 4,0 say " " color (colorvar)
1181 @ 4,0 say " " color (colorvar)

```

10/13/92	HWCOMP.SPR	08:13:57
<p>Author's Name</p> <p>Copyright (c) 1992 Company Name</p> <p>Address</p> <p>City, Zip</p> <p>Description: This program was automatically generated by GENSCRN.</p>		

# HWCOMP Setup Code - SECTION 1

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63

10/13/92	HMCOMP.SPR	00:13:57
<div style="display: flex; justify-content: space-between;"> <div style="width: 30%;"> <p><b>Author's Name</b></p> <p>Copyright (c) 1992 Company Name</p> <p>Address</p> <p>City, Zip</p> </div> <div style="width: 70%;"> <p><b>Description:</b> This program was automatically generated by GENSCRM.</p> </div> </div>		
<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">HMCOMP Setup Code - SECTION 1</div> <pre> region 1 parameter chascid, chascname if parameter() = 0 return endif  region 0 regional m.currares, m.talkstat, m.comstat  if set("TALK") = "ON" set talk off m.talkstat = "ON" else m.talkstat = "Off" endif comstat = set("COMPATIBLE") set compatible foxplus </pre> <div style="border: 1px solid black; padding: 5px; margin-top: 5px; text-align: center;">Window definitions</div> <pre> if not exist("w_hmcomp") define window w_hmcomp from int((arow()-20)/2), int((scol()-77)/2) ; to int((arow()-20)/2)+16, int((scol()-77)/2)+76 ; title "COMPUTE COST VALUE" ; nofloat ; noclose ; shadow ; double ; color scheme 1 endif </pre>		

```

67 *
68 *
69 *
70 *
71 *
72 #region 1
73 mlderr = on("ERROR")
74 msafe = set("safety")
75 set safety off
76 on error
77 do open
78 declare steparray[1]
79 m.step = 1
80
81 declare factarray[1]
82 m.fact = 1
83
84 declare phasearray[1]
85 m.phase = 1
86
87 m.sample = 200
88 store 1 to m.bbstep, m.bbfact, m.bbwidth, m.bbheight, m.bbphase
89 select hmtab
90 copy struc to hmcomp
91 use
92 select 0
93 use hmcomp
94 do get_hmarray
95
96 *
97 *
98 *
99 *
100 *
101 *
102 *
103
104 #region 1
105 if wvisible("w_hmcost")
106 activate window w_hmcost same
107 else
108 activate window w_hmcost noshow
109 endif
110 @ 1,34 get m.chmscid ;
111 size 1,3;
112 default 0;
113 picture "a";
114 disable
115 @ 1,38 get m.chmscname ;
116 size 1,15 ;
117 default " ";
118 picture "a";
119 disable
120 @ 3,2 get m.cstep ;
121 picture "a";
122 size 1,5;
123 default 0;
124 @ 2,21 get m.step ;
125 picture "a";
126 from steparray ;
127 size 3,42 ;
128 default 1;
129 color scheme 1, 2
130 @ 3,68 get m.bstep;
131 picture "a";
132 size 1,5 ;

```

## HMCOMP Screen Layout

```

133 default 0 ;
134 valid qpVhnr933()
135 a 6,2 get m.cfact ;
136 picture %g "m" ;
137 size 1,5 ;
138 default 0 ;
139 a 5,21 get m.fact ;
140 picture %g "m" ;
141 from factarray ;
142 size 3,42 ;
143 default 1 ;
144 c or scheme 1, 2
145 a 6,68 get m.bfact ;
146 picture %g "C" "m" ;
147 size 1,5 ;
148 default 0 ;
149 valid qpVhnr91uc()
150 a 9,2 get m.custep ;
151 picture %g "C" "m" ;
152 size 1,5 ;
153 default 0 ;
154 a 8,21 get m.wfact ;
155 picture %g "m" ;
156 from factarray ;
157 size 3,19 ;
158 default 1 ;
159 color scheme 1, 2
160 a 8,48 get m.bustep ;
161 picture %g "m" ;
162 from steparray ;
163 size 3,15 ;
164 default 1 ;
165 color scheme 1, 2
166 a 9,68 get m.bustep ;
167 picture %g "C" "m" ;
168 size 1,5 ;
169 default 0 ;
170 valid qpVhnr97qc()
171 a 12,2 get m.cphase ;
172 picture %g "C" "m" ;
173 size 1,5 ;
174 default 0 ;
175 a 11,21 get m.phase ;
176 picture %g "m" ;
177 from phasearray ;
178 size 3,42 ;
179 default 1 ;
180 color scheme 1, 2
181 a 12,68 get m.bphase ;
182 picture %g "C" "m" ;
183 size 1,5 ;
184 default 0 ;
185 valid qpVhnr931c()
186 a 16,3 get m.sample ;
187 size 1,12 ;
188 default 0 ;
189 picture %g "m" ;
190 valid qpVhnr93bm()
191 a 15,47 get m.scost ;
192 picture %g "RIM Yes" ;No" ;
193 size 1,10,0 ;
194 default 2 ;
195 a 17,26 get m.action ;
196 picture %g "H1 \<ok;\>@rowse;\<Cancel" ;
197 size 1,8,3 ;
198 default 1 ;

```

```

199 valid qpVhnr93kth()
200 a 9,40 say "at step:"
201 a 3,10 say "for Step:"
202 a 6,10 say "for Factor:"
203 a 9,10 say "for Factor:"
204 a 12,10 say "for Phase:"
205 a 1,25 say "Scenario:"
206 a 15,21 say "Total of Scenario cost:"
207 a 14,1 to 17,19
208 a 0,0 say "Estimated"
209 a 1,2 say "Cost"
210 a 0,65 say "Estimated"
211 a 1,66 say "Variance"
212 a 15,3 say "of iterations"
213
214 if not visible("w_hmcomp")
215 activate window w_hmcomp
216 endif
217
218 read cycle modal
219
220 release window w_hmcomp
221
222 #region 0
223 if m.talkstat = "ON"
224 set talk on
225 endif
226 if m.compatstat = "ON"
227 set compatible on
228 endif
229
230 *
231 *
232 *
233 *
234 *
235 *
236
237 #region 1
238 delfile("hmcomp")
239 delfile("hmtemp")
240 delfile("hmtab")
241 delfile("hmtest")
242 delfile("hmtmp")
243 set safety &unsafe
244 on error &mderr
245 return
246 *****
247 procedure open
248 *****
249
250 filesuccess = openfile("hmtab")
251 do cancel with filesuccess
252
253 filesuccess = openfile("hmtab")
254 do cancel with filesuccess
255
256 filesuccess = openfile("hmtab")
257 do cancel with filesuccess
258 set order to tag hmtab
259
260 filesuccess = openfile("hmtab")
261 do cancel with filesuccess
262 set order to tag hmtab
263
264 filesuccess = openfile("hmtab")

```

HMCOMP Cleanup Code

```

265 do cancel with filesuccess
266 set order to tag hmpid
267
268 filesuccess = openfile("hmc(")
269 do cancel with filesuccess
270
271 return
272
273 *****
274 function delfile
275 *****
276 parameter file
277 if used(file)
278 select (file)
279 use
280 delete file (file + ".dbf")
281 endif
282 return
283
284 *****
285 procedure cancel
286 *****
287 parameter success
288 if !m.success
289 warning("File doesn't Exist !!!!!", 2)
290 do cancel
291 endif
292 return
293
294 *****
295 procedure get hmparray
296 *****
297 private msel
298 msel = select()
299 select *;
300 from hmtab;
301 where hmcid = hmcscid;
302 into table hmtmp
303
304 select hmtmp
305 set relation to hmcid into hmc
306 if reccount() > 0
307 declare steptarray[reccount(), 2]
308 i = 0
309 j = 0
310
311 i = i + 1
312 steptarray[i, 1] = str(hmstep)
313 steptarray[i, 2] = reccount()
314 do get_table with hmtid, hmcid, hmpid
315 m.lcphase = alltrim(hmc.lcphase)
316 if ascscan(steptarray, lcphase) = 0
317 j = j + 1
318 declare phasearray[j, 2]
319 phasearray[j, 1] = alltrim(hmc.lcphase)
320 phasearray[j, 2] = hmcid
321 endif
322 endscan
323
324 if i > 1
325 declare steptarray[i+1, 2]
326 =all(steptarray, 1)
327 steptarray[i, 1] = "All Steps"
328 steptarray[i, 2] = 0
329 endif
330 if j > 1

```

```

331 declare phasearray[j + 1, 2]
332 =all(steptarray, 1)
333 phasearray[i, 1] = "All phases"
334 phasearray[i, 2] = 0
335 endif
336 set relation to && close relationship
337
338 select hmtcomp
339 if reccount() > 0
340 set relation to hmcid into hmc
341 i = 0
342
343 i = i + 1
344 factor = alltrim(hmc.factor)
345 if ascscan(factarray, factor) = 0
346 declare factarray[i, 2]
347 factarray[i, 1] = alltrim(hmc.factor)
348 factarray[i, 2] = hmcid
349 endif
350 endscan
351 asort = (factarray)
352 if i > 1
353 declare factarray[i+1, 2]
354 =all(factarray, 1)
355 factarray[i, 1] = "All factors"
356 factarray[i, 2] = 0
357 endif
358
359 endif
360 select (msel)
361 return
362
363 *****
364 procedure get table
365 *****
366 parameter mmatid, lcid, wpid
367
368 private msel
369 msel = select()
370 select *;
371 from hmtab;
372 where hmatid = m.matid and hmcid = m.lcid and hmpid = m.wpid;
373 into table temp
374
375 select hmtcomp
376 append from temp
377 select (msel)
378
379 return
380
381 *****
382 procedure compute
383 *****
384 private msel
385 msel = select()
386 sccount = 0
387 dimension stepttotal[1], facttotal[1], wfacttotal[1], phasetotal[1]
388
389 if m.cstep = 1
390 if steptarray[m.step, 2] = 0
391 select hmtmp
392 go top
393 dimension stepttotal[reccount()]
394 i = 0
395
396 i = i + 1

```

```

454 size = alen(phasearray,1)
455 dimension phasetotal[size-1]
456   for j = 2 to size
457     m.total = computphase(phasearray[j,2])
458     phasetotal[j - 1] = alltrim(phasearray[j,1]) + u + alltrim(
=> str(m.stotal,8,2))
459   endfor
460   -else
461     dimension phasetotal[1]
462     m.total = computphase(phasearray[m.phase,2])
463     phasetotal[1] = alltrim(phasearray[m.phase,1]) + u + alltrim(st
=> r(m.stotal,8,2))
464   endif
465   bootstrap = ""
466   nl = chr(10) + chr(13)
467   -do case
468     -case m.bstep = 1
469       select hntemp
470         go steparray(m.bbstep,2)
471         m.bttotal = bstep(hntemp,hmatid, hntemp.hmlcid, hntemp.hmupid, hnt
=> mp.pernum, hntemp.durnum, hntemp.qtynum)
472         bootstrap = "The bootstrap for step "+ alltrim(str(hntemp.hmatstep))
=> + ": " + nl + "Standard deviation - " + dp(m.bttotal,u,1) + space(10) +
=> "Mean - " + dp(m.bttotal, u,2)
473       -case m.bfact = 1
474         m.bttotal = bfact(factarray[m.bbfact,2])
475         bootstrap = "The bootstrap for factor "+ alltrim(factarray[m.bbfac
=> t,1]) + ": " + nl + "Standard deviation - " + dp(m.bttotal,u,1) + space(10) +
=> "Mean - " + dp(m.bttotal, u,2)
476       -case m.bwstep = 1
477         m.bttotal = bwstep(factarray[m.bbwfact,2], steparray[m.bbwstep,2])
478         bootstrap = "The bootstrap for factor "+ alltrim(factarray[m.bbwf
=> act,1]) + " within step " + alltrim(steparray[m.bbwstep,1]) + ": " + nl
=> + "Standard deviation - " + dp(m.bttotal,u,1) + space(10) + "Mean - " + dp
=> (m.bttotal, u,2)
479       -case m.bphase = 1
480         m.bttotal = bphase(phasearray[m.bbphase,2])
481         bootstrap = "The bootstrap for phase "+ alltrim(phasearray[m.bbpha
=> se,1]) + ": " + nl + "Standard deviation - " + dp(m.bttotal,u,1) + space(10) +
=> "Mean - " + dp(m.bttotal, u,2)
482       -endcase
483       do display with chmsname, scenttotal, steptotal, facttotal, wfacttotal,
=> phasetotal,bootstrap
484       select (mesl)
485       return
486     *****
487     procedure hmbrowse
488     *****
489     private mesl
490     mesl = select()
491     m.title = "SCENARIO: " + alltrim(upper(chmsname) )
492     select hntemp
493     set relation to hmatid into hmat
494     set relation to hmlcid into hmlc additive
495     set relation to hmupid into hmap additive
496     browse fields hmatop:h="STEP",hmat.hmatname:30:h="Material",;
=> hmlc.hmlc:30:h="Life cycle Phase",;
=> hmap.hmap:30:h="Working process",;
=> pernum:5:h="# of Employee",;
=> durnum:5:h="# of Quantity" nomodify normal title m.title
497

```



```

507 select (msel)
508 return
509
510 *****
511 function computstep
512 *****
513 parameter mat, lcpase, wp, an, sd, sq
514
515 private msel
516 msel = select()
517 select sum(hmcomp.wtaverage), 0000.00, hmcomp.hmetid, hmcomp.hmetprob,
518 => hmcomp.hmcfid,;
519
520 => nit;
521 from hmcomp;
522 where hmcomp.hmatid = m.mat;
523 and hmcomp.hmcid = m.lcpase;
524 and hmcomp.hmpid = m.wp;
525 group by hmcomp.hmetid, hmcomp.hmcfid, hmcomp.hmetprob;
526 into table test
527 m.sum = calculate(an, sd, sq)
528
529 select (msel)
530 return m.sum
531
532 *****
533 function computfact
534 *****
535 parameter m.factid
536
537 private msel
538 if parameter() = 0
539 return 0
540
541 msel = select()
542 select hntemp
543 go top
544 m.sum = 0
545
546 m.wpid = hntemp.hmpid
547 m.matid = hntemp.hmatid
548 m.lcid = hntemp.hlcid
549 select sum(hmcomp.wtaverage), 0000.00, hmcomp.hmetid, hmcomp.hmetpr
550 => ob,;
551 hmcomp.hmcfid, hmcomp.hmetid, hmcomp.hmetpr, hmcomp.hmetpr
552
553 from hmcomp;
554 where hmcomp.hmatid = m.matid;
555 and hmcomp.hmcid = m.lcid;
556 and hmcomp.hmpid = m.wpid;
557 and hmcomp.hmcid = m.factid;
558 group by hmcomp.hmetid, hmcomp.hmcfid, hmcomp.hmetid;
559 order by hmcomp.hmetid;
560 into table test
561 m.sum = m.sum + calculate(hntemp.pernum, hntemp.durnum, hntemp.qty
562 => num)
563
564 select (msel)
565 return m.sum
566
567 *****
568 function computfact
569 *****
570 parameter m.factid

```

```

568 private msel
569 if parameter() = 0
570 return 0
571
572 msel = select()
573 if steparray(m.wstep, 2) = 0
574 select hntemp
575 m.result = ""
576 go top
577
578 m.matid = hntemp.hmatid
579 m.lcid = hntemp.hlcid
580 m.wpid = hntemp.hmpid
581 select sum(hmcomp.wtaverage), 0000.00, hmcomp.hmetid, hmcomp.hmetpr
582 => tprob, hmcomp.hmcfid,;
583 hmcomp.hmcfid, hmcomp.hmetpr, hmcomp.hmetpr, hmcomp.hmetpr, hmcomp.hmetpr
584
585 from hmcomp;
586 where hmcomp.hmatid = m.matid;
587 and hmcomp.hmcid = m.lcid;
588 and hmcomp.hmpid = m.wpid;
589 and hmcomp.hmcid = m.factid;
590 group by hmcomp.hmetid, hmcomp.hmcfid, hmcomp.hmetid;
591 order by hmcomp.hmetid;
592 into table test
593 m.result = if(empty(m.result), "", m.result + u) + alltrim(str(hm
594 => temp.hntemp)) + ":" + alltrim(str(calculate(hntemp.pernum, hntemp.durnum, h
595 => mtemp.qtnum), 8, 2))
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626

```

```

627  _scan
628  _if hmcid = m.phaseid
629    m.matid = hntemp.hmatid
630    m.lcid = hntemp.halcid
631    m.wpid = hntemp.hmwpid
632    select sum(hmcomp.wtaverage), 0000.00, hmcomp.hmatid, hmcomp.hmcid
633    >= tprob, hmcomp.hmcid,
634    hmcomp.hmcfeid, hmcomp.perp, hmcomp.perd, hmcomp.perq, hmcomp.hmcid;
635    from hmcomp;
636    where hmcomp.hmatid = m.matid;
637    and hmcomp.halcid = m.lcid;
638    and hmcomp.hmwpid = m.wpid;
639    group by hmcomp.hmatid, hmcomp.hmcfeid, hmcomp.hmcid;
640    order by hmcomp.hmatid;
641    into table test
642    m.sum = m.sum + calculate(hntemp.pernum, hntemp.durnum, hntemp.
643    _endscan
644    select (msel)
645    return m.sum
646  _endscan
647  _if used('test')
648    _return 0
649  _endif
650  select test
651  t = 0
652  _if recount() > 0
653    _scan
654    s = 1
655    _if perp = 1
656      s = s * sn
657    _endif
658    _if perd = 1
659      s = s * sd
660    _endif
661    _if perq = 1
662      s = s * sq
663    _endif
664    s = sum_wtaver * s
665    replace exp_2 with s
666    _endscan
667  _endif
668  select sum(exp_2), hntestprob;
669  from test;
670  group by hntestid;
671  into cursor exp
672  _scan
673  _if recount() > 0
674    t = 0
675    _scan
676    _if empty(hntestprob)
677      hntestprob = 1/(recount()-1, .01)
678    _endif
679    t = t + (sum_exp_2 * hntestprob)
680  _endscan
681  _endif

```

```

690  _del('EXP')
691  select (msel)
692  return t
693  _endscan
694  _if empty(hntestprob)
695    hntestprob = 1/(recount()-1, .01)
696  _endif
697  parameter mat, lcpase, wp
698  msel = select()
699  select test
700  _scan
701  m.wt = selut(m.mat, m.lcpase, m.wp, test.hmatid, test.hmcid, tes
702  _endscan
703  replace sum_wtaver with m.wt
704  _endscan
705  select (msel)
706  return
707  _endscan
708  _if empty(hntestprob)
709    hntestprob = 1/(recount()-1, .01)
710  _endif
711  parameter mat, lcpase, wp, metid, cfid, cfeid
712  msel = select()
713  select hmcfeid, prob;
714  from hntab;
715  where hntab.hmatid = m.mat;
716  and hntab.halcid = m.lcpase;
717  and hntab.hmwpid = m.wp;
718  and hntab.hmatid = m.mat;
719  and hntab.hmcid = m.cfid;
720  and hntab.hmcfeid = m.cfeid;
721  into table test1
722  _endscan
723  select test1
724  m.randnum = rand()
725  m.wt = 0
726  m.start = 0
727  m.end = 0
728  _scan
729  m.end = 1/(empty(m.end), test1.prob, end + test1.prob)
730  _if m.randnum > m.start and m.randnum < m.end
731    m.wt = test1.hmcfeid
732  _endif
733  _endscan
734  m.start = m.start + test1.prob
735  _del('test1')
736  select (msel)
737  return m.wt
738  _endscan
739  _if empty(hntestprob)
740    hntestprob = 1/(recount()-1, .01)
741  _endif
742  function setupboot
743  _scan
744  _if steparray(m.step, 2) = 0
745    _copy(steparray, temparray)
746    _del(temparray, 1)
747    declare temparray[alen(temparray, 1) - 1, 2]
748    m.temp = beselect(temparray, "Select step: ")
749    _if empty(m.temp)
750      _errmsg("No Bootstrap computing...")
751    _endif
752    m.bstep = 0
753    m.pos = ascan(m.steparray, m.temp)
754    m.bstep = asubscript(m.steparray, m.pos, 1)

```

```

755 _endif
756 release temparray
757 _else
758 m.bbstep = m.step
759 _endif
760 _case m.bfact = 1
761 if factarray[m.fact,2] = 0
762 _copy(factarray, temparray)
763 declare temparray[1,2]
764 m.temp = bselect(temparray, "Select factor : ")
765 if empty(m.temp)
766 _errmsg("No Bootstrap computing...", 2)
767 m.bfact = 0
768 _else
769 m.pos = ascn(m.factarray, m.temp)
770 m.bbfact = subscript(m.factarray, m.pos, 1)
771 _endif
772 release temparray
773 _else
774 m.bbstep = m.step
775 _endif
776 _case m.bwstep = 1
777 if steparray[m.wstep,2] = 0 or factarray[m.wfact,2] = 0
778 _copy(steparray, temparray)
779 if temparray[1,2] = 0
780 _adel(temparray, 1)
781 declare temparray[alen(temparray, 1) - 1, 2]
782 _endif
783 _copy(steparray, temparray)
784 if temparray[1,2] = 0
785 _adel(temparray, 1)
786 declare temparray[alen(temparray, 1) - 1, 2]
787 _endif
788 m.temp = bfact(8temparray, 8temparray)
789 if empty(m.temp)
790 _errmsg("No Bootstrap computing...", 2)
791 m.bwstep = 0
792 _else
793 m.pos = ascn(factarray, dp(m.temp, u, 1))
794 m.bbfact = subscript(m.factarray, m.pos, 1)
795 m.pos = ascn(steparray, dp(m.temp, u, 2))
796 m.bbstep = subscript(m.steparray, m.pos, 1)
797 _endif
798 release temparray, temparray1
799 _else
800 m.bbwstep = m.step
801 m.bbfact = m.fact
802 _endif
803 _case m.bphase = 1
804 if phasearray[m.phase,2] = 0
805 _copy(phasearray, temparray)
806 _adel(temparray, 1)
807 declare temparray[alen(temparray, 1) - 1, 2]
808 m.temp = bselect(8temparray, "Select Life cycle phase:")
809 if empty(m.temp)
810 _errmsg("No Bootstrap computing...", 2)
811 m.bphase = 0
812 _else
813 m.pos = ascn(phasearray, m.temp)
814 m.bbfact = subscript(m.phasearray, m.pos, 1)
815 _endif
816 m.bphase = m.phase
817 _else
818 m.bphase = m.phase
819 _endif
820

```

```

821 _endif
822 _endcase
823 show gets
824 return
825 *****
826 function bstep
827 *****
828 parameter mat, lcpase, wp, sn, sd, sq
829 creat table boot (cost n(10,2))
830 m.boot = ""
831 select sum(hmcomp.waverage), 0000.00, hmcomp.hmetid, hmcomp.hmetprob
832 => , hmcomp.hmcfid, hmcomp.perp, hmcomp.perd, hmcomp.perq, hmcomp.hmu
833 => nit;
834 from hmcomp;
835 where hmcomp.hmetid = m.mat;
836 and hmcomp.hmcfid = m.lcpase;
837 and hmcomp.hmpid = m.wp;
838 group by hmcomp.hmetid, hmcomp.hmcfid, hmcomp.hmetid;
839 order by hmcomp.hmetid;
840 into table test
841
842 for i = 1 to m.sample
843 m.resetwt(m.mat, m.lcpase, m.wp)
844 m.sum = calculate(sn, sd, sq)
845 select boot
846 append blank
847 replace cost with m.sum
848 _endfor
849 calculate std(cost), avg(cost) to m.std, m.mean
850 m.boot = alltrim(str(m.std, 8, 4)) + u + alltrim(str(m.mean, 8, 2))
851 _del("boot")
852 return m.boot
853 *****
854 function bfact
855 *****
856 parameter m.factid
857 private msel
858 if parameter() = 0
859 _return 0
860 _endif
861 m.boot = ""
862 msel = select()
863 msel = select()
864 create table boot (cost n(10,2))
865 for i = 1 to m.sample
866 select hntemp
867 go top
868 m.sum = 0
869 _scan
870 m.wpid = hntemp.hmpid
871 m.matid = hntemp.hmetid
872 m.lcid = hntemp.hmcfid
873 select sum(hmcomp.waverage), 0000.00, hmcomp.hmetid, hmcomp.hme
874 => tprob,;
875 hmcomp.hmcfid, hmcomp.hmcfid, hmcomp.perp, hmcomp.perd,;
876 hmcomp.perq, hmcomp.hmu;
877 from hmcomp;
878 where hmcomp.hmetid = m.matid;
879 and hmcomp.hmcfid = m.lcid;
880 and hmcomp.hmpid = m.wpid;
881 and hmcomp.hmcfid = m.factid;
882 group by hmcomp.hmetid, hmcomp.hmcfid, hmcomp.hmetid;
883 order by hmcomp.hmetid;
884 into table test

```

```

884      =resetw(m.matid, m.lcid, m.wpid)
885      m.sum = m.sum + calculate(hatemp.pernum, hatemp.durnum, hatemp.
=> qtnum)
886      Lendscan
887      select boot
888      append blank
889      replace cost with m.sum
890      endfor
891      select boot
892      calculate std(cost), avg(cost) to m.std, m.mean
893      m.boot = alltrim(str(m.std,8,4)) + u + alltrim(str(m.mean,8,2))
894      =delete('boot')
895      select (mboot)
896      return m.boot
897
898      *****
899      function bustep
900      *****
901      parameter m.factid, m.stepid
902      private msel
903      if parameter() = 0
904      return 0
905      Lendif
906      m.boot = ""
907      msel = select()
908      create table boot (cost n(10,2))
909      select hatemp
910      go (m.stepid)
911      m.matid = hatemp.matid
912      m.lcid = hatemp.lcid
913      m.wpid = hatemp.wpid
914      select sum(hatemp.utaverage), 0000.00, hatemp.matid, hatemp.pernum,
=> hatemp.hatempid;
915      hatemp.hatempid;
916      Lendif
917      from hatemp;
918      where hatemp.matid = m.matid;
919      and hatemp.lcid = m.lcid;
920      and hatemp.wpid = m.wpid;
921      group by hatemp.matid, hatemp.hatempid;
922      order by hatemp.matid;
923      into table test
924      Lendif
925      =resetw(m.matid, m.lcid, m.wpid)
926      m.sum = calculate(hatemp.pernum, hatemp.durnum, hatemp.qtnum)
927      select boot
928      append blank
929      replace cost with m.sum
930      endfor
931      select boot
932      calculate std(cost), avg(cost) to m.std, m.mean
933      m.boot = alltrim(str(m.std,8,4)) + u + alltrim(str(m.mean,8,2))
934      =delete('boot')
935      select (mboot)
936      return m.boot
937
938      *****
939      function bphase
940      *****
941      parameter m.phaseid
942      private m.sum, msel
943      if parameter() = 0

```

```

947      Lendif
948      m.boot = ""
949      msel = select()
950      create table boot (cost n(10,2))
951      for i = 1 to m.sample
952      select hatemp
953      go top
954      m.sum = 0
955      Lendscan
956      if hatempid = m.phaseid
957      m.matid = hatemp.matid
958      m.lcid = hatemp.lcid
959      m.wpid = hatemp.wpid
960      select sum(hatemp.utaverage), 0000.00, hatemp.matid, hatemp.
=> hatemp.hatempid;
961      hatemp.hatempid;
962      Lendif
963      from hatemp;
964      where hatemp.matid = m.matid;
965      and hatemp.lcid = m.lcid;
966      and hatemp.wpid = m.wpid;
967      group by hatemp.matid, hatemp.hatempid;
968      order by hatemp.matid;
969      into table test
970      Lendif
971      =resetw(m.matid, m.lcid, m.wpid)
972      m.sum = m.sum + calculate(hatemp.pernum, hatemp.durnum, hatemp.
=> hatemp.qtnum)
973      hatemp.hatempid;
974      Lendif
975      select boot
976      append blank
977      replace cost with m.sum
978      endfor
979      select boot
980      calculate std(cost), avg(cost) to m.std, m.mean
981      m.boot = alltrim(str(m.std,8,4)) + u + alltrim(str(m.mean,8,2))
982      =delete('boot')
983      select (mboot)
984      return m.boot
985
986      *
987      *
988      *
989      *
990      *
991      *
992      *
993      *
994      *
995      *
996      *
997      *
998      *
999      *
1000      function qpvh983 22 m.bstep VALID
1001      $region 1
1002      if bstep = 1
1003      bfact = 0
1004      bustep = 0
1005      bphase = 0
1006      show gets
1007      Lendif
1008
1009      *

```

Function Origin: m.bstep VALID

From Screen: HMCMP, Record Number: 6

Variable: m.bstep

Called By: VALID Clause

Object Type: Check Box

Snippet Number: 1



REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE Sept 1992		3. REPORT TYPE AND DATE COVERED Final-Oct 91 through Sept 92
4. TITLE AND SUBTITLE Hazardous Material Life-Cycle Cost Model Technical Manual			5. FUNDING NUMBERS Program Element: REIMB Work Unit Number:  NAVFAC.WR.1026W	
6. AUTHOR(S) H. Ly and D. M. Pearsall				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Health Research Center P. O. Box 85122 San Diego, CA 92186-5122			8. PERFORMING ORGANIZATION Report No. 92-19	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Naval Medical Research and Development Command National Naval Medical Center Building 1, Tower 2 Bethesda, MD 20889-5606			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT  Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words)  This technical manual contains the information on the program code, data elements, file structures needed to maintain the Hazardous Material Life-Cycle Cost Module. This documentation was created using the FOXDOC Version 2.0 program.				
14. SUBJECT TERMS Life-cycle cost                      Technical Model                                  Hazardous materials Cost-benefit analysis              Hazardous waste			15. NUMBER OF PAGES 157	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT Unlimited	